华章数学译丛

34



Applied Combinatorics (Second Edition)

应用组合数学

（原书第2版）

（美）Fred S. Roberts 拉特格大学 Barry Tesman 狄克森学院 著

冯速 译





（原书第2版）

应用组合数学

本书写作方法非常出色，第2版保持了前一版的高质量，并进行了大量更新。书中内容叙述非常翔实，便于学生理解，例子讲解生动并富有启发性，而且所涉及的应用范围之广更是罕见。

——John Elwin，圣迭戈州立大学

本书介绍组合数学的基本知识和应用，涉及计算机科学、生物学、化学、心理学及基因工程等前沿学科中的最新应用，应用层面非常宽泛。本书布局精巧、内容翔实，对题材的讨论深入浅出，简明扼要，包含了很多高级的组合数学技术与方法。全书分为四个部分：第一部分介绍组合数学的基本工具，第二部分介绍计数问题，第三部分讲述组合数学求解中的存在问题，第四部分讨论优化问题。

本书第1版曾被国外多所大学采纳为教材，这一版根据最新技术发展进行了大量修改，书中包含大量出色的实例和练习，可作为高等院校数学专业和计算机科学专业“组合数学”课程的教材。

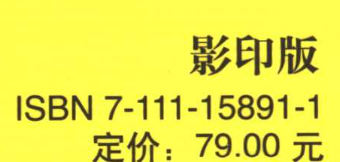
作者简介



Fred S. Roberts 美国拉特格大学数学系教授，研究方向包括数学模型在社会学、行为学、生物学、环境科学以及传媒和交通方面的应用，图论与组合数学，测度论等。



Barry Tesman 于美国拉特格大学获得数学专业博士学位，目前为狄克森学院数学与计算机科学系副教授。他的研究方向包括图论、组合数学和测度论。





Applied Combinatorics

(Second Edition)



应用组合数学

影印版

ISBN 7-111-15891-1

定价：79.00 元

华章网站 http://www.hzbook.com

网上购书：www.china-pub.com

投稿热线：(010) 88379604

购书热线：(010) 68995259, 68995264

读者信箱：hzjsj@hzbook.com



HZ BOOKS

华章图书

封面设计·杨宇梅

ISBN 978-7-111-20934-8

定价：69.00 元

华章数学译丛

34

0157
17
2007

Applied Combinatorics (Second Edition)

应用组合数学

（原书第2版）

（美）Fred S. Roberts 拉特格大学 Barry Tesman 狄克森学院 著

冯速 译



机械工业出版社
China Machine Press

译 者 序

本书以组合数学的实际应用为背景，非常全面地讲述了组合数学的基本知识和基本问题，以及这些知识在计算机科学、密码学、政治经济学、生物学、医学、遗传学等各个领域的实际应用.

本书分四个部分. 第一部分讲述组合数学的基本工具，而其余三个部分分别讲述组合数学的三个基本问题：计数问题、存在问题和优化问题. 各章以实际应用为例，引出该章的主要课题，然后对前后关联的知识点逐步展开讨论，在讨论中又提出更多的示例，力图让读者掌握所讲述的思想并灵活运用相关的方法. 另外，各节给出很多相关的练习，使读者能够检验所学知识，同时引入新概念和新应用，为读者灵活运用书中的组合技术提供素材. 因此，本书的一大特点就是包含大量的实际例子及练习，其中许多例子和练习都来自于最新的文献. 这些例子和练习前后呼应、反复出现，形成若干系列，可以看出经过了作者的精心筛选.

本书的另一个特点是强调算法. 当今计算机的快速发展和各种算法的开发是促进组合数学快速发展和广泛应用的主要动力之一，而组合数学的快速发展同样促进了计算机理论的发展及计算机在各个领域的广泛应用. 所以，组合数学与计算机科学之间已经形成了密不可分的关系，组合数学是计算机科学——特别是计算机算法的基础. 现在，组合数学已经成为多数大学计算机专业研究生的必修课程.

本书的第三个特点是，不仅给出了绝大多数结果的严密证明，而且不拘泥于通常教科书所用的概念和结果，对一些概念和结果进行了扩展.

本书可以作为计算机专业和数学专业高年级本科生或研究生的教材，也可以作为相关科研人员的参考书. 不满足于"离散数学"课程所学内容的学生可以通过本书继续自学，而本书各章后的参考文献则为希望继续深入学习、了解科研前沿及应用前沿的读者提供了参考.

在翻译本书时，我们遇到了许多不同专业的术语. 我们力图使术语简单、易懂，一些术语没有采用数学专业的说法，而是采用计算机科学专业的说法. 对于中文特别难以反映其本质内容的术语，我们采取了不译的做法. 由于书中涉及应用的多样性和译者的水平所限，难免有不当之处，敬请读者批评指正.

译 者
于北京师范大学

前 言

本书第 2 版的面世距初版时间已有 20 年之久. 新版进行了大幅度修订，加入 200 多页㊀新材料，许多章节有了重大变更，并增加了大量新的例子与练习. 但是，本书的宗旨没有改变. 下面三段话引自第 1 版前言，这些文字至今仍是适用的.

> 当代数学发展最快的领域可能就是组合数学了. 它之所以发展快速，一个主要原因是它在计算机科学、通信、交通运输、遗传学、实验设计和排程等方面的广泛应用. 因此，本书将从应用的观点向读者介绍组合数学的工具.
>
> 组合数学的发展是同计算机的发展齐头并进的. 当今的高速计算机使得各领域中实际组合问题的求解成为可能，而这些问题在不久以前还是无法解决的. 这种情况增强了研究组合问题求解方法的重要性. 与此同时，计算机科学的发展本身又带来了大量具有挑战性的组合问题. 因此，我们很难把组合数学和计算技术割裂开来. 读者将会看到本书中对计算技术的强调，我们会经常采用计算机科学中的大量应用实例，频繁讨论算法，等等. 另一方面，我们认为：组合数学在大量学科中有着广阔的应用前景，我们强调的重点是应用的多样性，而不限于某一方面.
>
> 本书选取的数学题材中，许多是快速发展的组合数学教科书中相对而言称为标准的内容. 另一部分题材的选择，或者来自当前的研究文献，或者由于其学科应用价值而被选用. 我们相信，本书在对应用的广泛讨论方面是与众不同的. 书中有若干完整的小节专门介绍这样的应用，例如开关函数、使用酶揭示未知的 RNA 链，信息检索中的搜索与排序，错误校正码的构造，化学成分的计算，选举中的势力预测以及斐波那契数的应用. 还有一些小节介绍与卷积有关的递推的应用、欧拉链的应用以及生成函数的应用等，这在许多教材中是没有的.

第 2 版中的新内容

本书之所以受到欢迎，大部分原因在于它引用了最新文献和各种实际应用. 推动着组合数学发展的应用仍然在不断扩展，尤其是在自然科学和社会科学中. 在第 2 版中，许多新应用更是来自计算机科学和生物学. 在增加应用题材的同时，我们也增加了某些新的主题，删去了一些特殊的内容，改变了组织结构，并对例子、练习和参考文献也做了改进与更新.

下面说明第 2 版中所做的某些主要修改.

第 1 章：增加了有关列表着色的新材料，扩充了对州立法会议日程安排的讨论. 列表着色问题在书中许多地方还会出现.

第 2 章：在第 1 版中，2.16 节仅限于讨论生成排列的算法方法，第 2 版现在补充了大量材料，并分成几个小节. 2.18 节介绍“好算法”的概念和 NP 完全问题，进行了大幅度改写和更新. 此外，还添加了有关“鸽巢原理”的新章节. 讨论拉姆齐理论的小节是由第 1 版 8.1 节的材料和 8.2 节的部分材料合成的. 另外，我们还增加了讨论排列之间的倒位距离和进化论生物学中的突变问题的内容.

第 3 章：在 3.3 节中增加了图着色问题的一小节，着重讨论图着色的扩展，例如多重着色和

㊀ 这是指英文原书。——编辑注

T着色. 这类着色问题是由现实应用引起的，例如移动电话问题、交通灯切换和信道分配. 此外，在本章中我们还增加了物种进化树重构的内容. 对于第1版第8章有关拉姆齐理论的许多材料中没有写进第2章的部分，已改写为新的一节.

第4章：这一章是全新的. 内容包括二元关系的定义及其与有向图的联系，用有向图和关系引入的顺序以及对线性序和弱序的处理，偏序、线性扩展和维数，链和可比较图，格，布尔代数，开关函数和门网络. 该章紧密结合应用来讨论，其中涉及信息论、效用理论、搜索与排序以及诸如开关函数这样一些较早的应用. 这一章包含的某些应用在组合数学的文献中没有受到广泛关注，如优先选择、搜索引擎、杂交顺序和心理物理量表法等. 基于该章中的概念的许多例子也出现在以后各章中. 第4章涉及的范围可以一直延续到第11章之后.

第5章：在第1版中这是第4章. 前面几章引入的许多新概念与例子在这一章又重新出现，例如第4章的弱序和第3章的列表着色.

第6章：这是第1版中的第5章. 该章增加了有关DNA序列对位的新材料，作为排列的"转置平均值"的内容.

第7章：这是第1版中的第6章. 该章增加了有关密码技术和整数因子分解的新素材(在本书后面还要提到).

第1版原第8章：这一章已经被删去，其中的重要材料加进第2章，而其余的材料已纳入书中其他章节.

第8章：这是第1版中的第7章. 该章中新增加了一些例子，这些例子以第4章中的概念(如弱序)为基础. 同时，该章中还增加了一小节，讨论图的自同构，并在该章多处地方提到.

第9章：该章增加了讨论正交阵列和密码技术的一节，内容涉及识别码和秘密分享. 与此同时，该章还讨论了模算术与RSA密码系统之间的联系，以及秘密分享应用的一个可求解设计的内容. 此外，该章关于"组群测试"的新内容涉及几个应用，其中包括次品鉴别、疾病筛查、基因组图谱测绘和卫星通信.

第10章：有关"合意解码"的小节是新增的，这同寻找分子序列中的蛋白质有关，同时加入了涉及光盘错误校正码的内容. 关于"判读"DNA产生蛋白质的材料也是新增加的.

第11章：在11.2节中增加了讨论单行线问题的几小节. 这些新的内容涉及有关广场取向和栅格的最新结果，分别反映不同城市类型的需要. 该章还增加了关于测试巨大图的连通性问题的内容，这类问题出现在与电信业务量和网上数据有关的最新应用中. 此外，该章还包括关于DNA杂交顺序的内容.

第12章：这一章用若干新的例子解释概念，包括天花疫苗接种、音响系统和石油钻井的例子. 我们写了新的一节讨论稳定婚姻问题及其许多最新的应用，包括医院实习医生分配等. 第1版第13章中有关最大权匹配的内容已移至该章中.

第13章：我们引入有关门格定理的新内容. 该章还收入许多新的实例，讨论诸如建筑物疏散、聚类和数据挖掘以及分布式计算等问题.

保留的特点

尽管第2版对第1版做了重大修改，但仍然突出了使本书具有独特价值的那些特点.

- 强调来自不同领域中的应用，把这些应用作为本来的主题而不只是一些孤立的例子，并且从最新文献中寻找应用.
- 许多例子在书中反复出现，特别是那些与新旧主题都有联系的例子.
- 强调通过不同的练习求解问题，用这些练习检测对常规概念掌握的程度，引入新的概念

和应用，或者试图对读者提出应用现有组合数学技术的挑战. 本书依然坚持这样一个原则，即学习组合数学的最佳方式是通过求解问题学习，实际上这也是学习任何一门数学的最佳方式.

- 把不同难度的主题组合在一起，再加以详细的注释，使本书可以在各种不同层次的课程中使用.
- 本书的组织方式允许读者按各种不同的顺序使用其中的题材，这一方面反映出组合数学中主题的某种独立性，同时这样灵活的方式又使各主题相得益彰，便于读者学习.

本书的组织方式

本书分为四个部分. 第一部分包括第 2、3、4 章，介绍组合数学中使用的基本工具及其应用，其中引入基本计数规则以及图论与关系中使用的工具. 其余三部分围绕组合数学的三个基本问题(计数问题、存在问题和优化问题)组织. 这些问题在第 1 章中讨论. 本书第二部分涉及处理计数问题的更高级工具，其中包括生成函数、递推关系、容斥原理以及波利亚理论. 第三部分涉及解的存在问题，其中讨论组合设计、编码理论和图论中的特殊问题. 这也是有关图论和网络的连续三章(第 11 章至第 13 章)的开始，并作为图算法的导论. 第四部分讨论组合优化，通过对图和网络的连续研究说明各种基本思想. 这部分从过渡性的匹配与覆盖一章开始，其中又以存在问题开头，以优化问题结束. 第四部分最后讨论图和网络的优化问题. 将本书分为四个部分的做法带有某种主观性，因为书中的许多主题涉及组合数学的若干方面，例如某些主题既包括存在问题，又包括优化问题. 然而，对于庞大的现代组合学题材，把本书分成四个部分看起来又是很合理的.

预备知识

本书可以在不同层次的课程中使用. 本书大部分内容是针对主修和非主修数学和计算机科学专业的低年级和高年级学生编写的. 本书也适用于具有足够数学基础的大学二年级学生(书中已清晰地指明了在入门课程中可以省略的题材). 另一方面，书中包含足够的材料，能以一种快速的进度用于具有难度的研究生课程. 作为本科课程的教材，本书已在拉特格大学中使用过，选课的大部分学生来自数学系和计算机科学系，其余的学生来自商学、经济学、生物学和心理学系. 在狄克森学院，书中的材料已在主修数学的低年级或高年级的学生中使用过. 这些课程以及本书的预备知识(包括熟悉函数和集合)，至少可以从微积分课程中获得. 第 5 章和第 6 章要用到无穷序列和无穷级数(其实第 6 章大部分只用到无穷序列的基本性质，并不需要极限的概念). 这里无需了解微积分中的其他传统的主题. 但是，从微积分这样的课程中获得数学上的素质训练则是很必要的. 此外，读者需要了解某些线性代数的工具，特别是熟悉矩阵运算. 我们还假定读者了解数学归纳法(有一些教师会在课程的早期阶段复习数学归纳法，并快速复习集合的知识). 书中少数可选读的几节需要用到本书所述以外的概率知识. 另有几节介绍近世代数的内容，例如群和有限域. 上述这些节是自足的，但是对于没有足够基础的学生而言，这样的进度可能过快.

算法

本书在很多部分把重点放在算法上. 由于组合数学日益用于研究精确和有效求解复杂问题的过程，以及组合数学和计算机科学如此紧密的结合，这样做是不可避免的. 我们的目标是引入算法的概念并介绍一些重要的算法实例. 我们在讨论大部分的算法时，都采用一种相对形式化

的方式. 这种方式很少去揭示算法的概念以及如何描述算法. 主要目标在于提出算法的基本思想，而不试图以最简洁或适合计算机的形式描述算法. 有人不同意用这种方法介绍算法. 我们的观点则是不能用组合数学的课程代替算法的学习. 计算机科学系的学生需要学习一门单独的算法课程，其中应包括对实现所述算法的数据结构的讨论. 然而，学习组合数学的全体学生需要接受算法的思想，具备算法的思维方式，这种思维方式对于本学科而言是重要的也是基本的. 我们认识到，我们对如何介绍算法的折中方案并不能使所有的人满意. 但是这里应当指出，对于具有计算机背景的学生来说，应完成书中那些有趣的且重要的练习，把其中非形式化的算法转换成更精确的计算机算法甚至是计算机程序.

例子和应用所起的作用

书中的各种应用起了很大作用，使本书在众多的组合数学书籍中独树一帜. 作者建议教师对应用进行挑选，或把它们作为课外阅读材料. 书中的许多应用是用例子的形式介绍的，并在全书中重复展现. 无论对教师或是学生而言，都不必成为书中不同实例和各个小节所述的应用领域的专家. 应用和例子多半是自足的，倘若不是这种情况，也很容易通过因特网上的相关搜索获得对应用的理解.

组合数学同计算机科学之间的联系极其重要，在此无需特别强调.

本书中取材于生物科学的例子非常重要. 我们把重点放在这样的例子上是基于下述观察：生物科学和数学科学之间的联系正在以极快的速度加强，数学和计算机科学的方法在现代生物学中扮演了且仍在扮演着重要的角色，例如，在“人类基因工程”和疾病传播建模中. 让数学工作者认识到专业领域需要像组合数学这样的数学方法已变得越来越重要. 此外，已经有越来越多的学校开设数学和生物学相结合的交叉课程.

数学与社会科学之间的联系程度也在迅速增加，尽管其进展速度不如数学同生物科学的联系程度. 现在，计算机科学的工具和数学建模方法已应用于处理日益复杂的社会科学问题. 所以，我们介绍了出现在社会科学方面的种种应用，重点是决策和投票问题.

数学证明

作出证明是数学区别于其他学科的一个重要方面. 组合数学可能是一种非常好的学习数学证明的途径，通过它可以向学生介绍数学证明的概念，并教会他们如何写出完善的证明. 有些学校以组合数学作为数学证明课程的入门课程. 但这不是我们写这本书的目的. 虽然使用本书的教师应把证明包括在内，但我们却倾向于把证明处理成略带非形式化的，而非把重点放在如何写证明上. 我们将书中许多比较难的证明加上了星号，表示是可选的.

练习

本书的练习处于主角的地位. 练习用于检验对常规概念掌握的程度，引入新的概念和应用，并试图作为对读者运用书中既有的组合技术提出的挑战. 组合数学的本质在于，越是通晓它，越能应付种种难题，这也是大部分数学的本质. 在本书中，我们尽量收入各种实用性和理论性的练习以及具有不同难度的练习.

在不同情况下使用本书的方法

本书可适用于不同层次的不同课程. 我们已在几门课程中使用书中的材料，特别是在名为“组合数学”的一学期课程和名为“应用图论”的一学期课程中使用. 对低年级和高年级讲授的组合

数学课程，用第 1、2、3、5、6、7、9、10 章的大部分材料，略去正文中由脚注指明的各节(通常为证明). 在拉特格大学一门快进度的一年级研究生课程中，Fred Roberts 把课程的重点放在证明上，此外还在课程中融入许多可选的章节，同时采用第 8 章或第 12 章的材料. 无论在本科生或研究生的课程中，教师都应该用第 8 章或第 11 章以及第 12 章和第 13 章的部分内容代替第 9 章和第 10 章. 在不单独开设图论课程的情况下，特别推荐把第 11 章包括在内. 类似地，如果不开设运筹学课程，特别推荐把第 13 章的部分内容包括在内. 在拉特格大学，对本科生和研究生分别开设的课程，讲授第 11 章至第 13 章的大部分材料.

其他的一学期或暑期课程应按照本书的材料设计，因为大多数章是相对独立的(请参考下面的讨论). 在拉特格大学，应用图论课程是围绕第 3 章和第 11 章讲授的，并从书中的其余部分(第 4、12 和 13 章)及别的地方补充图论的题材(需要快速讲授 2.1 节至 2.7 节，或者还有 2.18 节). 第 3、11、12、13 章也适于在一门介绍图算法的课程或者名为“图与网络”的课程中讲授. 全书非常适合作为现代组合数学及其应用引论的一学年教材. 对于学习过组合数学的人来说，若要将重点放在组合数学的应用上，可以把书中有关应用的小节和例子包括在内.

本书可以用于大学二年级的一学期或暑期课程. 这样的课程应包含第 1 章至第 3 章，跳过第 4 章和第 5 章，然后选用第 6 章的 6.1 节和 6.2 节，以及第 7 章和第 11 章的部分内容，带有星号的各节和大多数证明应予省略. 其他的题材可由教师酌情增加.

主题之间的相关性和使用本书题材的顺序

在组织任何课程时，希望教师了解对题材相对独立性的说明. 在讲授组合数学引论的主题时，没有被普遍接受的顺序，对这样的引论应包含哪些主题也没有一致的看法. 我们在写本书时力求采用这样一种方法，使各章的内容相对独立，并能按不同的顺序讲授.

第 2 章是本书的基础部分，介绍全书用到的基本计数规则. 第 3 章只包含足以进入到主题的图论知识，它的重点是用以说明第 2 章提出的基本计数规则的那些图论题材. 第 3 章引入的概念在全书各处都会涉及，尤其是在第 4、11、12 和 13 章. 教师可以把本书用于一学期或暑期的组合数学课程，不讲授第 3 章. 但是，我们建议至少应把图着色(3.3 节和 3.4 节)的材料包含在内. 除第 3 章外，主要依赖关系是：第 4 章依赖第 3 章；第 6 章 6.2 节之后的内容依赖第 5 章；第 7 章引用第 3 章和第 6 章中的几个例子；第 11 章至第 13 章依赖第 3 章；10.5 节依赖第 9 章；第 12 章中提到的几个概念在 13.3.8 节中会用到.

处于快速发展时期的组合数学

最后应该强调，组合数学是一门快速发展的学科，它的方法正在迅速建立，它的应用正在迅速扩展. 本书介绍的许多主题同研究工作的前沿紧密相关. 它是一门可以把新手很快带到前沿领域的典型学科. 我们尽力收集组合数学及其应用的参考文献，以便使有兴趣的读者能更深入地钻研本书所讨论的主题.

致谢

作者之一 Fred Roberts 于 1976 年开始第 1 版的写作，当时为拉特格大学组合数学的本科课程写了一部分手稿. 经过几年时间的修改和增补，本书正式用于组合数学课程和前述的其他课程，Barry Tesman 也这样使用本书. 同时还有很多人也同样把本书作为他们的教材，并反馈了大量意见，这使作者受益匪浅. 作者特别要感谢下列使用者提出的非常有价值的评论：Midge Cozzens，在东北大学使用第 1 版的草稿；Fred Hoffman，在佛罗里达大西洋大学使用；Doug

West，在普林斯顿大学使用；Garth Isaak，在 Lehigh 大学使用第 2 版的草稿；Buck McMorris，在伊利诺伊理工学院使用第 2 版的草稿.

我们特别感谢现在和过去的学生们，他们在本书两个版本的准备阶段以多种方式提供了帮助，包括校对、核对练习、检查各种错误，此外他们还提出了各种宝贵的意见. Fred Roberts 对 Midge Cozzens、Shelly Leibowitz、Bob Opsut、Arundhati Ray-Chaudhuri、Sam Rosenbaum 和 Jeff Steif 等人的帮助表示感谢. Barry Tesman 对 Kathy Clawson 和 John Costango 的帮助表示感谢.

我们收到许多人对本书的评论. 我们要特别感谢以下人士，他们在审阅第 1 版的各个阶段和其他时间曾做出极其有价值的评论：John Cozzens、Paul Duvall、Marty Golumbic、Fred Hoffman、Steve Maurer、Ronald Mullin、Robert Tarjan、Tom Trotter 以及 Alan Tucker. 在准备第 2 版的时候，我们收到了 Steve Maurer 对第 1 版提供的非常有用的评论. Jeff Dinitz 对第 9 章和第 10 章的草稿提出了具体的意见. 关于第 2 版，我们收到审阅者很有价值的评论，他们是：Edward Allen、Martin Billik、John Elwin、Rodney W. Forcade、Kendra Killpatrick、Joachim Rosenthal、Sung-Yell Song、Vladimir Tonchev 和 Cun-Quan Zhang. 尽管我们收到大量的意见和评论，但书中的疏误仍然在所难免，我们会对此负责.

本书第 1 版的成书过程，几经录入、复制和剪贴(剪刀加浆糊). 在这期间，Fred Roberts 得到 Lynn Braun、Carol Brouillard、Mary Anne Jablonski、Kathy King、Annette Roselli 和 Dotty Westgate 的莫大帮助. 由于出版业务的显著变化，第 2 版又由 Barry Tesman 进行录入和剪贴(电子编辑)等工作. 由于第 1 版没有电子副本，其扫描任务是由 Barry Tesman 以前的学生 Jennifer Becker 完成的. 在此，Barry Tesman 对她承担这一艰巨任务表示感谢，同时还要感谢狄克森学院数学与计算机科学系的人员提供的支持和所做的贡献.

两位作者在这里还要感谢家人的支持. 写作意味着要占用大部分的家庭时间：为了校对要停用电话，为了写作要取消旅游，为了某处改进要牺牲休息时间. 我们的家人对此给予极大的帮助和理解. Fred Roberts 感谢已故双亲 Louis 和 Frances Roberts 的关爱与支持；感谢已故岳母 Lily Marcus 在技术上和其他方面提供的帮助；感谢妻子 Helen 的帮助，虽然她看上去就像是一位因丈夫写书而“独守空房的女子”，但她的帮助不只限于一如既往的支持、劝导和鼓励，同时她也是本书其中一章的合作者，在书中引入了她在教学中使用的各种题材和例子，这些材料遍及书中各处；最后还要感谢 Sarah 和 David，在写第 1 版时，孩子们的贡献就是保持稳定良好的情绪，更为重要的是在成年以后，他们以其他方式对本书做出了贡献，比如 Sarah 介绍的公共卫生的理念目前已反映在本书中，David 对计算机科学的许多方面做出了解释，对此在本书的一处脚注中对他表示感谢. Barry Tesman 感谢双亲 Shirley 和 Harvey Tesman 的关爱和支持；感谢妻子 Johanna，在这项任务中，她是默默的同伴，而在过去的 20 年岁月中，她却是活跃的伴侣和朋友；最后要感谢 Emma 和 Lucy，因为他们是最可爱的.

Fred S. Roberts
froberts@dimacs. rutgers. edu
Barry Tesman
tesman@dickinson. edu

记　号

集合论记号

$\cup$	并集	$\varnothing$	空集
$\cap$	交集	$\{\cdots\}$	集合……
$\subseteq$	子集(包含于)	$\{\cdots:\cdots\}$	所有满足……的集合……
$\subsetneq$	真子集	A^c	集合A的补
$\nsubseteq$	不是子集	$A-B$	集合A与B的差，$A\cap B^c$
$\supseteq$	包含(超集)	$\lvert A\rvert$	集合A的势，A中的元素数量
$\in$	……的元素		
$\notin$	非……的元素		

逻辑记号

$\sim$	非
$\rightarrow$	蕴涵
$\leftrightarrow$	当且仅当(等价)
iff	当且仅当

其他常用记号

$\lfloor x\rfloor$	小于或等于x的最大整数	$\approx$	近似等于
$\lceil x\rceil$	大于或等于x的最小整数	$\equiv$	与……同余
$f\circ g$	两个函数f和g的合成	A^{T}	矩阵A的转置
$f(A)$	函数f下集合A的像，即$\{f(a): a\in A\}$	Π	积
(a, b)	由满足$a<c<b$的所有实数c组成的开区间	Σ	和
$[a, b]$	由满足$a\leqslant c\leqslant b$的所有实数c组成的闭区间	$\int$	积分
		Re	实数集合

本书介绍组合数学的基本知识，以及这些知识在计算机科学、生物学、医学、遗传学等各个领域的实际应用. 全书分为四个部分：第一部分介绍组合数学的基本工具，第二部分介绍计数问题，第三部分讲述组合数学求解中的存在问题，第四部分讨论优化问题.

本书布局精巧、内容翔实，讨论深入浅出，简明扼要，可作为高等院校数学专业和计算机科学专业"组合数学"课程的教材，也可以作为相关科研人员的参考书.

Simplified Chinese edition copyright © 2007 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Applied Combinatorics*, *Second Edition* (ISBN 0-13-079603-4) by Fred S. Roberts and Barry Tesman, Copyright © 2005, 1984.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Prentice Hall.

本书封面贴有 Pearson Education(培生教育出版集团)激光防伪标签，无标签者不得销售.

版权所有，侵权必究.

本书法律顾问　北京市展达律师事务所

本书版权登记号：图字：01-2005-0522

图书在版编目(CIP)数据

应用组合数学(原书第2版)/(美)罗伯茨(Roberts, F. S.)等著；冯速译. —北京：机械工业出版社，2007.5

(华章数学译丛)

书名原文：Applied Combinatorics, Second Edition

ISBN 978-7-111-20934-8

Ⅰ. 应…　Ⅱ. ①罗…②冯…　Ⅲ. 组合数学　Ⅳ. O157

中国版本图书馆 CIP 数据核字(2007)第035697号

机械工业出版社(北京市西城区百万庄大街22号　邮政编码　100037)

责任编辑：迟振春

北京牛山世兴印刷厂印刷 · 新华书店北京发行所发行

2007年5月第1版第1次印刷

186mm×240mm · 36.75印张

定价：69.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线：(010)68326294

目　录

第二部分 计数问题

第三部分　存在问题

第四部分 组合优化

第1章 什么是组合数学

1.1 组合数学的三个问题

现代数学中发展最快的领域也许就是组合数学了．组合数学的研究对象是排列、模式、设计、配置、调度、关联和布局．在当今世界中，活跃在各个领域的人们发现解决具有组合数学性质的问题的必要性．计算机科学家为了给复杂的语句编码，需要考虑数字和开关的*模式*(pattern)．商店经理着手进行工具或工作场所的人员*配置*(assignment)．农业专家为不同的田地*分配*(assign)试验苗．电器工程师筛选电路*布局*(configuration)．银行家研究电子资金转账的*模式*(pattern)，而空间科学家研究把信息传送到遥远的卫星的*模式*(pattern)．工业工程师为使生产高效考虑产品的*调度*(schedule)和厂房的*布局*(configuration)．大学的教务官*安排*(arrange)教室以满足时间段和学生之间的*调度*(schedule)．化学家考虑不同原子和分子的可能*关联*(connection)以及分子中原子的*排列*(arrangement)．运输管理人员*安排*(arrange)公共汽车或飞机的*调度*(schedule)．语言学家考虑使用未知字母的单词*排列*(arrangement)．遗传学家考虑把基的*排列*(arrangement)组合成DNA链、RNA链等．统计学家考虑为实验*设计*(design)替代方案．

这些都是组合数学的基本问题．这些基本问题是*存在问题*(existence problem)、*计数问题*(counting problem)和*优化问题*(optimization problem)．存在问题研究的是：是否至少存在一种特定种类的排列？计数问题研究的是：存在多少种排列？优化问题所关心的是：在各种可能的排列中，选择对应于某个标准来说最好的排列．我们用若干例子来说明这三个问题．

【例1.1 实验设计】 考虑测试五种药物对人体药效的实验设计．假设这些药物编号为1，2，3，4，5．我们可以挑选五个人，并给每一个人不同的药物．

可是，特定人可能对特定药物产生过敏反应，或对该药效是免疫的．因此，我们有可能得到偏差很大的结果．更有效的方法应该是在连续的一段期间，例如5天，给每个人每一种药物．表1.1给出这一实验的一种可能安排．这一安排有什么错误之处吗？其一，一种药物在这一周摄取的日期可能对结果产生影响(比如在周一早晨还没有醒酒的人可能对周一的药物没有正常的反应)．其二，较早时吃的药可能对晚些时候吃的药的性能产生影响．因此，让每个人按相同顺序吃这些药物可能导致结果的偏差．

表1.1 药物实验设计

		日期				
		M	Tu	W	Th	F
对象	A	1	2	3	4	5
	B	1	2	3	4	5
	C	1	2	3	4	5
	D	1	2	3	4	5
	E	1	2	3	4	5

注：其中，每行对应于给定的实验对象，每列对应于给定的日期，各项表示某一天某个人吃了某种药．

解决这些问题的一个方法就是简单地要求不能有两个人在同一天吃相同的药物．那么这一实验的设计就形成一个5×5的表格，它的每一项都是数字1，2，3，4，5中的一个，且每一行的项完全不同，每一列的项也完全不同．这是一个特殊的模式．对这一药物实验的设计者来说，关键问题是：存在这样的设计吗？这就是组合数学的存在问题． ■

下面从更一般的角度提出这一问题．我们定义*拉丁方*(Latin square)[㊀]是以数字1，2，3，…，n为项的$n\times n$表格，而且不存在数字在同一行或同一列出现多于1次的情况．同时，拉丁方要求

㊀ 术语"拉丁方"来自于这样的事实：表格中的元素通常是用拉丁字母表中的字母表示的．

每一个数字在每一行和每一列内正好出现一次．一个典型的存在问题是：存在 2×2 的拉丁方吗？答案是存在；表 1.2 给出一个这样的方格．类似地，可以考虑是否存在 3×3 的拉丁方．这一次答案也是肯定的；表 1.3 给出一个 3×3 拉丁方．

我们的特定问题是：是否存在 5×5 的拉丁方？表 1.4 表明答案是肯定的．（对于每一个 n，是否存在 $n\times n$ 的拉丁方？答案是已知的，我们把它留给读者．）

2

表 1.2　一个 2×2 拉丁方

1	2
2	1

表 1.3　一个 3×3 拉丁方

1	2	3
2	3	1
3	1	2

表 1.4　一个 5×5 拉丁方

1	2	3	4	5
2	3	4	5	1
3	4	5	1	2
4	5	1	2	3
5	1	2	3	4

注意，拉丁方仍然不是可能引发顺序效应问题的完全解．为了避免任意可能的顺序效应，应该有足够多的对象使得这五种药物的每一种可能顺序都可以得到测试．那么存在多少这样的顺序呢？这就是计数问题(counting problem)，即组合数学中的第二个基本问题．结果是存在 $5!=120$ 个这样的顺序，2.3 节会给出这一数值的求解方法．因此，应该有 120 个对象．如果只有五个对象可用，那么应该随机选取所使用的拉丁方来设法避免顺序效应．这时，有多少种可选的 5×5 拉丁方呢？我们将在 6.1.3 节学习这一计数问题．

正如这一粗略的讨论所提示的那样，实验设计的问题是促进组合数学发展的一个主要因素[⊖]．我们将在第 9 章详细讨论实验设计问题．

【例 1.2 位串和二进制代码】 位(bit)或二进制数字(binary digit)是 0 或 1．位串(bit string)定义为诸如 0001、1101 或者 1010 这样的位的序列．位串是现代计算机中信息的主要载体．位串用于编码详细的指令，它被翻译成计算机中各开关上的开-关指令的序列．一个符号集合的二进制代码(binary code)(或二进制块代码(binary block code))给每一个符号指定一个不同的位串．考虑字母表中 26 个字母的二进制代码．一个典型的代码是莫尔斯(Morse)码，它的更传统形式是用点表示 0，用短线表示 1．下面是一些常用字母的莫尔斯码：

O：111，A：01，K：101，C：1010.

如果只能使用一个或两个位组成位串，那么我们能够编码所有 26 个字母吗？答案是否定的，因为只有如下所示 6 个可能的位串：

0，1，00，01，10，11.

注意，为了回答所提出的问题，我们必须计数(count)可能的排列数量．这是求解计数问题的例子．在这种情况下，我们通过枚举(enumerating)列出所有可能的排列来计数．通常，这太乏味或太浪费时间，而我们将要开发不通过枚举的计数捷径．现在，三位或更短的位串是否足够编码字母表中所有 26 个字母呢？答案同样是否定的．简单的枚举表明只存在 14 个这样的位串．（你能列举出它们吗?）然而，四位或更短的位串就足够了．（这样的位串有多少个?）的确，莫尔斯码只使用四个或更少的符号．它不使用所有可能的位串．（为什么?）在 2.1 节研究遗传代码时，我们将遇到一个非常类似的计数问题．DNA 链给所需要的基本遗传信息编码，以确定称为蛋白质的氨基酸长链．我们将设法解释编码一个氨基酸需要 DNA 链中多长的片段．代码不仅仅出现在遗传或现代计算机通信中，本书的其他部分也出现一些代码．例如，在第 10 章，将研究错误校

3

⊖ 参看 Herzberg 和 Stanton[1984].

正码，我们利用这些代码从遥远的航天探测器接收信息或者向其发送信息、发射导弹等. ■

【例 1.3 天然气管道的最佳设计】 天然气在管道里的流动依赖于管道的直径、长度、管道两端的压力、温度以及天然气的各种性质等. 海上天然气管道系统的设计涉及诸多问题，例如，如何确定各种接合点或连接点处的管道直径大小，以保证构建及运作的总成本最低. 解决这一问题的一个标准做法是使用"工程化判断"来挑出合理的管道尺寸，并且假设最佳的情况. 乍看起来，没有达到更好效果的可能. 例如，一个有 40 个连接点且每个连接点有 7 种可能管道尺寸的网络系统，会生成 7^{40} 个可能的网络，正如我们将在 2.1 节中看到的那样. 而 7^{40} 是一个非常大的数字. 我们的问题是从这 7^{40} 种可能中找到成本最低的网络. 这是组合问题的第三个基本问题——优化问题(optimization problem)的例子，在这样的问题中，我们设法寻找最优(最好、最大或最小等)的设计、模式或排列.

应该指出的是，伴随着计算机的发展，解决组合优化问题已有了很大的进展. 几年前在计算机上似乎还不可能求解的问题今天已可能求解. 因此，计算机的发展已成为组合优化领域快速发展的主要推动力. 然而，计算机所能做的事情也有局限. 我们将在下文中看到.

在理论上，任何有限问题都可以通过考虑所有可能性而得到解决. 然而，在上面所说的特定问题中，通过列举所有可能管道网络的可能性将花费多长时间呢？为了对此有所了解，请注意 7^{40} 近似于 6×10^{33}，即 6 后面跟着 33 个 0. 这是一个巨大的数字. 的确，即使是一台能够在 1 秒内分析出 10 亿(10^9)种不同管道网络的计算机(每十亿分之一秒一个网络)，也将花费大约 $1.9\times10^{17}=$ 190 000 000 000 000 000 年才能分析出所有 7^{40} 种可能的管道网络[⊖].

4

现代组合数学关心更多的是开发解决存在问题、计数问题和优化问题的程序或算法(algorithm). 从实际的角度看，在计算机科学中，分析解决一个问题的算法需要花费多长时间或需要多大存储容量是非常重要的问题. 在机器上着手处理一个计算(例如尝试所有可能性)之前，我们希望知道这一计算是否能够在合理的时间内，使用这台机器的可用存储容量完成. 我们将在 2.4 节和 2.18 节中讨论关于计算复杂度的问题.

对于上面讨论的管道问题，考虑所有可能情况似乎不是一个容易解决的问题，即使使用当今最高速的计算机也是如此. 计算速度的任何可预测改进都不可能给这一计算问题带来实质性的改进. 然而，一个简单的程序能够给出一个方法，这一方法仅用大约 $7\times40=280$ 步就可以完成寻找最优网络的工作，而不是使用 7^{40} 步. 墨西哥湾的工程使用了这一程序，节省了巨额资金. 参见 Frank Frisch[1970]、Kleitman[1976]、Rothfarb 等[1970]和 Zadeh[1973]. 这是组合优化技术发挥威力的一个例子. ■

【例 1.4 立法委员会会议安排】 州立法委员会一周举行一次例行会议. 在安排会议时间时，立法会议议长助理必须细心，不能使有共同成员的两个委员会在同一时间召开会议. 假设只有三个可用的会议时间：星期二、星期三和星期四的上午，必须安排会议时间的委员会有财政委员会、环境委员会、健康委员会、运输委员会、教育委员会和住房委员会. 假设表 1.5给出有共同成员的委员会的情况. 表示表 1.5 信息的一个便利方法是绘制一个图，在这个图中，使用点表示委员会，而且当且仅当两个点所对应的委员会有共同成员时，这两个点由一条无向的线连接. 其结果图称为图(graph).

⊖ 因为一年大约有 3.15×10^7 秒，所以一年内可以分析出 $3.15\times10^7\times10^9$ 或 3.15×10^{16} 个网络. 那么检查 6×10^{33} 个网络所需要的年数是

$$\frac{6\times10^{33}}{3.15\times10^{16}}\approx1.9\times10^{17}.$$

表 1.5 委员会中的共同成员的关系

	财政	环境	健康	运输	教育	住房
财政	0	0	0	0	0	1
环境	0	0	1	0	1	0
健康	0	1	0	1	1	1
运输	0	0	1	0	0	1
教育	0	1	1	0	0	1
住房	1	0	1	1	1	0

注：如果委员会 i 和 j 有共同成员，则 i，j 项是 1，否则是 0.（为了方便，对角线上的项取作 0.）

图 1.1 给出按此方法使用表 1.5 的数据所得到的图. 这种图有很多应用，例如在计算机科学、运筹学、电机工程、生态学、政策与决策学以及社会科学等各个领域. 我们将在第 3 章和第 11 章及其他章节讨论图及其应用.

第一个问题是：给定三个可用会议时间，我们能否寻找到委员

5

会会议时间的一种分配，使得所有成员都不在同一时间出席两个会议？这是一个存在问题. 利用所画出的图，我们希望给每一个点指定一个会议时间，使得如果两个点被线连接，那么这两个点得到不同的会议时间. 我们能够找到这样的一种分配吗？经过一些分析之后，本例的答案是肯定的. 一种可行的分配是这样的：住房委员会和环境委员会的会议在星期二举行，教育委员会和运输委员会的会议在星期三举行，而财政委员会和健康委员会的会议在星期四举行.

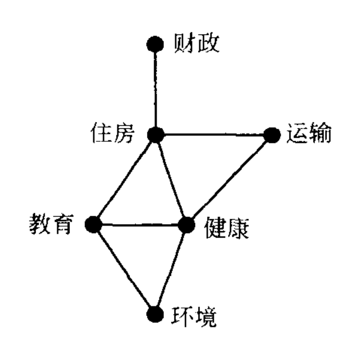


图 1.1 根据表 1.5 的数据得到的图

大学期末考试或班会时间的调度问题、工厂中的工作调度问题以及其他调度问题都类似于上面的会议时间安排问题. 我们将在第 3 章探讨这一问题. 在第 3 章中我们把这些问题看成是图着色(graph coloring)问题，并把本例中的会议时间看成是相应的“颜色”.

如果每一个委员会的主席提交一份可接受会议时间表，那么这一问题就会变得更现实. 于是我们会问，是否有一种委员会会议时间的分配，使得每一个委员会的会议时间都被安排在一个可接受的时间，而没有委员会的成员同时出席两个会议？例如，假设对于运输委员会来说可接受的会议时间是星期二和星期四，对于教育委员会来说可接受的会议时间是星期三，而其他委员会可接受三天中的任意一天. 那么，不难证明这一问题的解不存在.（参见练习 13.）于是我们

6

对这一存在问题给出否定的答案. 这就是称为*列表着色问题*(list-coloring problem)的调度问题的例子，即使用可接受列表中的颜色为指定颜色的图着色问题(在上面的情况下，颜色代表“一周的日期”). 我们将在例 3.22 中重新讨论这一问题. 列表着色近年来得到了广泛的研究，有关近期的研究，请参见 Alon[1993]以及 Kratochvil，Tuza 和 Voigt[1999].

表 1.6 首选会议时间

委员会	财政	环境	健康	运输	教育	住房
主席的首选时间	星期二	星期四	星期四	星期二	星期二	星期三

接下来我们也许会问：假设每一位委员会主席都明确给出他的首选会议时间，满足我们最初要求(如果存在这样的安排)并使得最大数量的委员会主席得到首选会议时间的会议时间分配是什么？这是一个优化问题. 考虑应该如何回答这一问题. 假设表 1.6 给出每一位委员会主席的首选会议时间. 解决这一优化问题的一个方法就是简单地列出所有满足原始条件的可能会议时间安排，并对每一个会议时间安排计数有多少委员会主席得到了他们的首选会议时间. 正如前

面所发现的那样，在实现组合问题的方法之前，我们希望对这种方法所消耗的时间有所了解. 需要分析多少种可能性呢？这是一个计数问题，可以通过枚举法解决. 从图 1.1 容易看到，住房、教育和健康委员会必须得到不同的会议时间. (它们之间都有一条连线.)类似地，运输委员会所得到的会议时间必须不同于住房委员会和健康委员会. (为什么?)因此，因为只有三个会议时间是可用的，所以运输委员会必须与教育委员会在同一时间举行会议. 类似地，环境委员会必须与住房委员会在同一时间举行会议. 最后，财政委员会不能与住房委员会在同一时间举行会议，因此也不能与环境委员会在同一时间举行会议，但是财政委员会可以与其他任何委员会同时举行会议. 因此，只存在如下所示的两种可能的会议模式.

模式 1. 运输委员会和教育委员会在一个时间举行会议，环境委员会和住房委员会在另一个时间举行会议，而财政委员会和健康委员会在第三时间举行会议.

模式 2. 运输、教育和财政委员会在一个时间举行会议，环境委员会和住房委员会在另一个时间举行会议，而健康委员会在第三时间举行会议.

表 1.7 给出了会议时间的所有可能分配情况，总共有 12 种. 这一计数问题通过列举所有可能性而得到解决. (在 3.4.1 节中，我们用另外一种方法解决这一计数问题.)根据例 1.3，显然枚举法不是总可以用来解决组合问题. 事实上，如果存在更多的委员会和更多的可能会议时间，那么上面所讨论的问题将使我们束手无策. 7

表 1.7 会议时间的可能分配

分配号码	星期二	星期三	星期四	得到首选会议时间的委员会主席的数量
1	运输 教育	环境 住房	财政 健康	4
2	运输 教育	财政 健康	环境 住房	3
3	环境 住房	运输 教育	财政 健康	1
4	环境 住房	财政 健康	运输 教育	0
5	财政 健康	运输 教育	环境 住房	2
6	财政 健康	环境 住房	运输 教育	2
7	运输 教育 财政	环境 住房	健康	5
8	运输 教育 财政	健康	环境 住房	4
9	环境 住房	运输 教育 财政	健康	1

（续）

分配号码	星期二	星期三	星期四	得到首选会议时间的委员会主席的数量
10	环境 住房	健康	运输 教育 财政	0
11	健康	运输 教育 财政	环境 住房	1
12	健康	环境 住房	运输 教育 财政	1

在我们的例子中，枚举法是成功的，我们能够很容易地解决这一优化问题．表1.7给出在每一种分配之下，得到首选会议时间的委员会主席的数量．显然，从该表可以看到，7号分配是最佳分配．这里，只有环境委员会主席没有得到他的首选会议时间．对于有关国家立法委员会的会议时间分配的详尽参考，可以参见 Bodin 和 Friedman[1971]；对于周期性(例如一周等)重复发生的其他调度问题，可以参考 Ahuja，Magnanti 和 Orlin[1993]、Baker[1976]、Bartholdi，Orlin 和 Ratliff[1980]、Chrétienne[2000]、Crama 等[2000]、Karp 和 Orlin[1981]、Orlin[1982]以及 Tucker[1975]；对于各种劳动力调度算法的研究，可以参见 Brucker[1998]、Kovalëv 等[1989]、Tien 和 Kamiyama[1982]. ■

本书是围绕着组合数学的三个基本问题而组织内容的．它分为四个部分：在第2章～第4章的介绍部分之后，其余三个部分研究计数问题(第5章～第8章)、存在问题(第9章～第11章)和优化问题(第12章～第13章)三个问题.

1.2 组合数学的历史和应用㊀

1.1节中所讨论的四个例子展示了组合数学所关心的一些问题，它们选自各领域，以说明组合数学在现代的广泛应用.

尽管组合数学在现代得到最大的发展，但它却是数学的一个古老分支．根据传说，大禹(大约公元前2200年)在一个神龟的背上发现了一个幻方．(幻方(magic square)是一个数字方阵，其每行、每列以及两条对角线上的各数字和都相同．表1.8给出一个这样的方阵．读者也许希望找到一个不同的3×3幻方.)

表1.8 幻方

4	9	2
3	5	7
8	1	6

公元前1100年，中国人就已经知道了排列．大约公元前300年，欧拉知道了当$n=2$时$(a+b)^n$的二项式展开．在匿名希伯来人写于公元200～500年之间的著作《创世书》(Sefer Yetzirah)中，发现了n个元素集合的排列数量公式的运用．这一公式本身至少在2500年前就已经知道了．公元1100年，Rabbi Ibn Ezra 就知道从n个事物中一次取r个事物的组合的数量的公式，即二项式系数．自那之后不久，中国人、印度人和阿拉伯人开始在著作中提及原始形式的二项式系数.

8～9

在近代，17世纪的学者帕斯卡和费马研究了与博弈相关的组合问题，还提出了概率的概念．(事实上著名的帕斯卡三角形早在1303年就为中国的朱世杰提出.)帕斯卡和费马的工作奠定了概率论的基础；18世纪，拉普拉斯使用有利情形定义了概率．也是在18世纪，结合著名的哥尼斯

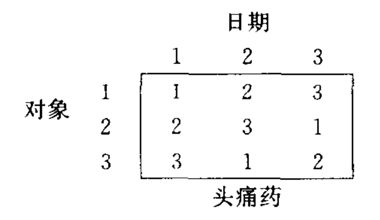
㊀ 对于组合数学历史的更详细讨论参见 Biggs，Lloyd 和 Wilson[1995]或者 David[1962]．对于图论的历史参见 Biggs，Lloyd 和 Wilson[1976].

堡桥问题，欧拉发明了图论，而伯努利出版了第一本展示组合方法的书《猜度术》(Ars Conjectandi). 在 18 世纪和 19 世纪，哈密顿和其他人把组合数学应用于拼图和游戏的研究中. 在 19 世纪，基尔霍夫发展了电网的图论方法，而凯莱发展了有机化学研究中的枚举技术. 在现代，组合数学技术已在计算机科学、运输、信息处理、工业规划、电机工程、实验设计、取样、编码、遗传学、政治学等其他重要领域有着广泛而重要的应用. 在本书中，我们总是取身边的应用，并且提醒读者它们不仅得益于数学技术的发展而且它们还促进着这些技术的持续发展.

练习

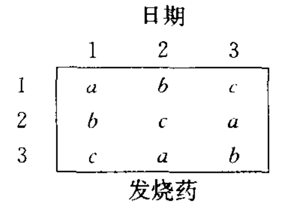
1. 寻找一个 4×4 拉丁方.
2. 寻找所有可能的 3×3 拉丁方.
3. 描述如何寻找一个 $n\times n$ 拉丁方.
4. (Liu[1972])假设我们要同时测试两种类型的药物，如头痛药和发烧药. 在这种情况下，我们也许尝试设计一个实验，在这一实验中，使用拉丁方测试每一种类型的药物. 然而，如果可能的话，我们还要确定测试的头痛药和发烧药的所有组合. 例如，表 1.9 给出当我们有 3 种头痛药和 3 种发烧药时的两种拉丁方设计. 在表 1.9 中还给出第三个方格，它的 i，j 项是前两个方格的 i，j 项. 我们要求这第三个方格的每一项都不相同. 在表 1.9 中这是不成立的.
   (a)对于有 3 种头痛药和 3 种发烧药的例子，寻找组合方格具有所希望性质的例子.
   (b)寻找另外一个有 4 种头痛药和 4 种发烧药的例子.（在第 9 章，将讨论对于有 6 种头痛药和 6 种发烧药的情况，这是不可能的. 这一存在问题没有解.）*注意*：如果你开始于一个头痛药的拉丁方设计，且不能找到一个发烧药的拉丁方设计使得组合方格具有所希望的性质，那么你应该开始于一个不同的头痛药设计.

表 1.9　测试头痛药 1，2，3 的拉丁方设计、测试发烧药 a，b，c 的拉丁方设计，以及这两种设计的组合



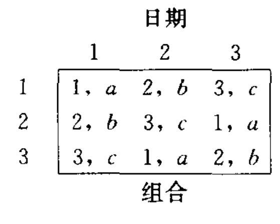
对象 \ 日期	1	2	3
1	1	2	3
2	2	3	1
3	3	1	2

头痛药



日期	1	2	3
1	a	b	c
2	b	c	a
3	c	a	b

发烧药



日期	1	2	3
1	1，a	2，b	3，c
2	2，b	3，c	1，a
3	3，c	1，a	2，b

组合

注：第三个方格的 i，j 项是前两个拉丁方中的 i，j 项中的头痛药和发烧药.

5. 通过枚举法证明存在 14 种长度至多为 3 的位串.
6. 使用枚举法寻找长度至多为 4 的位串的数量.
7. 假设我们要使用符号为 0、1 或 −1 的串来构建字母表中的 26 个字母的*三进制代码*(trinary code).
   (a)你能使用长度至多为 2 的位串编码所有 26 个字母吗？使用枚举法回答这一问题.
   (b)使用长度正好为 3 的位串时情况又如何？

10

8. DNA 分子中呈现的遗传代码是由四个字母 T，C，A，G 的符号串所组成的，我们将在 2.1 节中讨论这一代码. 通过枚举法确定使用这些字母并使其长度为 3 或小于 3 的不同的代码字或位串的数量.
9. 假设在天然气管道网络的设计中，我们有两种可能的管道尺寸，小号(S)和大号(L). 如果有 4 个可能的连接点，那么枚举出所有可能的管道网络.（典型的一种管道网络可以缩写为 LSLL，其中第 i 个字母表示第 i 个管道的尺寸.）
10. 在例 1.3 中，假设一台计算机在 1 秒钟内可以分析 1000 亿种不同的管道网络，比文中假设的速度提高了 100 倍. 这会对我们的结论带来什么明显的不同吗？为什么？（在给出你的答案时做一下计算.）

11. 表 1.10 和表 1.11 给出大学里若干课程在课程表中重叠的数据.
    (a)把表 1.10 转换成如例 1.4 中的图.
    (b)对于表 1.11 重复(a).

表 1.10 重叠数据

	英语	微积分	历史	物理
英语	0	1	0	0
微积分	1	0	1	1
历史	0	1	0	1
物理	0	1	1	0

注：当第 i 门课程与第 j 门课程有同一学员时 i，j 项为 1，否则为 0.

11

表 1.11 更多的重叠数据

	英语	微积分	历史	物理	经济学
英语	0	1	0	0	0
微积分	1	0	1	1	1
历史	0	1	0	1	1
物理	0	1	1	0	1
经济学	0	1	1	1	0

注：当第 i 门课程与第 j 门课程有同一学员时 i，j 项为 1，否则为 0.

12. (a)假设表 1.10 中所考虑的课程的期末考试只有两种可能. 是否存在一种期末考试时间安排，使得有同一学员的任意两门课程有不同的考试时间？如果存在，那么找出一种这样的时间安排；如果不存在，为什么？
    (b)假设存在三种可能的期末考试时间，对于表 1.10 重复(a).
    (c)假设存在三种可能的期末考试时间，对于表 1.11 重复(a).
    (d)假设存在四种可能的期末考试时间，对于表 1.11 重复(a).
13. 假设对于表 1.5 的情况，运输委员会的可接受会议时间是星期二和星期四，而教育委员会的可接受会议时间为星期三，其他所有委员会的可接受会议时间为星期二、星期三和星期四. 证明不存在可能的会议时间安排.
14. (a)假设对于表 1.10 的情况，每一门课程的可接受考试时间安排如表 1.12 所示. 另外假设每一门课程的考试必须在可接受时间进行，回答练习 12(b)的问题.
    (b)假设对于表 1.11 的情况，每一门课程的可接受考试时间如表 1.13 所示. 另外假设每一门课程的考试必须在可接受时间进行，回答练习 12(d)的问题.

表 1.12 可接受考试时间

课程	英语	微积分	历史	物理
可接受考试时间	星期四上午	星期三上午 星期四上午	星期二上午 星期三上午	星期二上午 星期三上午

表 1.13 更多可接受考试时间

课程	英语	微积分	历史	物理	经济学
可接受考试时间	星期三上午	星期二上午 星期三上午	星期二上午 星期三上午	星期二上午 星期四上午	星期一上午 星期三上午

15. 假设有三种可能的期末考试时间，星期二、星期三和星期四的上午. 假设表 1.10 所列出的这些课程的每一位教师要求星期二上午为期末考试的首选时间. 如果存在这样的考试时间安排的话，那么什么样的安排可以使得最大数量的教师得到首选时间？

12

参考文献

AHUJA, R. K., MAGNANTI, T. L., and ORLIN, J. B., *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, NJ, 1993.

ALON, N., "Restricted Colorings of Graphs," in K. Walker (ed.), *Surveys in Combinatorics*, Proceedings 14th British Combinatorial Conference, London Math. Soc. Lecture Note Series, Vol. 187, Cambridge University Press, Cambridge, 1993, 1–33.

BAKER, K. R., "Workforce Allocation in Cyclical Scheduling Problems," *Oper. Res. Quart., 27* (1976), 155–167.

BARTHOLDI, J. J., III, ORLIN, J. B., and RATLIFF, H. D., "Cyclic Scheduling via Integer Programs with Circular Ones," *Oper. Res., 28* (1980), 1074–1085.

BIGGS, N. L., LLOYD, E. K., and WILSON, R. J., *Graph Theory 1736–1936*, Oxford University Press, London, 1976.

BIGGS, N. L., LLOYD, E. K., and WILSON, R. J., "The History of Combinatorics," in R. L. Graham, M. Grötschel, and L. Lovász (eds.), *Handbook of Combinatorics*, Elsevier, Amsterdam, 1995, 2163–2198.

BODIN, L. D., and FRIEDMAN, A. J., "Scheduling of Committees for the New York State Assembly," Tech. Report USE No. 71–9, Urban Science and Engineering, State University of New York, Stony Brook, NY, 1971.

BRUCKER, P., *Scheduling Algorithms*, Springer-Verlag, Berlin, 1998.

CHRÉTIENNE, P., "On Graham's Bound for Cyclic Scheduling," *Parallel Comput., 26* (2000), 1163–1174.

CRAMA, Y., KATS, V., VAN DE KLUNDERT, J., and LEVNER, E., "Cyclic Scheduling in Robotic Flowshops," *Ann. Oper. Res., 96* (2000), 97–124.

DAVID, F. N., *Games, Gods, and Gambling*, Hafner Press, New York, 1962. (Reprinted by Dover, New York, 1998.)

FRANK, H., and FRISCH, I. T., "Network Analysis," *Sci. Amer., 223* (1970), 94–103.

HERZBERG, A. M., and STANTON, R. G., "The Relation Between Combinatorics and the Statistical Design of Experiments," *J. Combin. Inform. System Sci., 9* (1984), 217–232.

KARP, R. M., and ORLIN, J. B., "Parametric Shortest Path Algorithms with an Application to Cyclic Staffing," *Discrete Appl. Math., 3* (1981), 37–45.

KLEITMAN, D. J., "Comments on the First Two Days' Sessions and a Brief Description of a Gas Pipeline Network Construction Problem," in F. S. Roberts (ed.), *Energy: Mathematics and Models*, SIAM, Philadelphia, 1976, 239–252.

KOVALËV, M. Ya., SHAFRANSKIJ, Ya. M., STRUSEVICH, V. A., TANAEV, V. S., and TUZIKOV, A. V., "Approximation Scheduling Algorithms: A Survey," *Optimization, 20* (1989), 859–878.

KRATOCHVÍL, J., TUZA, Z., and VOIGT, M., "New Trends in the Theory of Graph Colorings: Choosability and List Coloring," in R. L. Graham, J. Kratochvíl, J. Nešetřil, and F. S. Roberts (eds.), *Contemporary Trends in Discrete Mathematics*, DIMACS Series, Vol. 49, American Mathematical Society, Providence, RI, 1999, 183–197. [13]

LIU, C. L., *Topics in Combinatorial Mathematics*, Mathematical Association of America, Washington, DC, 1972.

ORLIN, J. B., "Minimizing the Number of Vehicles to Meet a Fixed Periodic Schedule: An Application of Periodic Posets," *Oper. Res., 30* (1982), 760–776.

ROTHFARB, B., FRANK, H., ROSENBAUM, D. M., STEIGLITZ, K., and KLEITMAN, D. J., "Optimal Design of Offshore Natural-Gas Pipeline Systems," *Oper. Res., 18* (1970), 992–1020.

TIEN, J. M., and KAMIYAMA, A., "On Manpower Scheduling Algorithms," SIAM *Rev., 24* (1982), 275–287.

TUCKER, A. C., "Coloring a Family of Circular Arcs," SIAM *J. Appl. Math., 29* (1975), 493-502.

ZADEH, N., "Construction of Efficient Tree Networks: The Pipeline Problem," *Networks, 3* (1973), 1–32. [14]



第一部分　组合数学的基本工具

第 2 章　基本计数规则[⊖]

2.1　乘法规则

组合数学的基础由若干基本计数规则组成. 我们把这些规则概括到本章中. 对这些规则已经很熟的读者也许希望快速地复习一下它们. 本章还介绍在证明特定类型的排列或模式存在时广泛使用的工具. 在阅读本章时，已熟悉计数规则的读者也许希望重点了解不太熟悉的计数规则的各种应用，其中很多内容都将在以后的章节中重复使用.

【例 2.1 位串和二进制代码(重探例 1.2)】 我们回到二进制代码的例子(例 1.2)，重新考虑如果正好有 2 位，那么字母表中有多少个字母可以被编码. 让我们通过画出一个树图来得到答案. 图 2.1 是我们的树图. 正如前面所提到的那样，存在四种可能的 2 位的位串. 读者将会发现第一位存在 2 种选择，对于其中的每一种选择，第二位都有 2 种选择，所以共有4 种可能，即2×2. ■

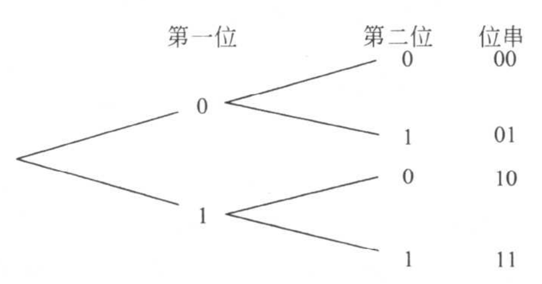


图 2.1　计数长度为 2 的位串数量的树图

【例 2.2 DNA】 一个有机体的所有遗传信息就是它的基因组(genome). 把基因考虑成一个长脱氧核糖核酸(DNA)分子是很便利的. (基因组实际上就是由表示各个染色体的 DNA 片段组成的.)DNA(或染色体)是由称为核苷的构件的串组成的. 基因组的大小可以用核苷的总数量来表示. 因为事实上 DNA 是一个双链结构，其中的两个链是由特定的基之间的(一个基是核苷的三个亚成分之一)配对组合到一起的. 因此，基因组的大小通常是由基对(bp)表示的. 核苷中的每一个基是四种可能的化学物质中的一个：胸腺嘧啶 T，胞嘧啶 C，腺嘌呤 A，鸟嘌呤 G. 基序列编码特定的遗传信息. 特别地，它确定了称为蛋白质的氨基酸长链. 有 20 种基本的氨基酸. DNA 分子中的基序列将编码一种这样的氨基酸. DNA 分子串必须有多长才有足够多可能的基来编码 20 种不同的氨基酸呢? 例如，2-元素的 DNA 序列能否编码 20 种不同的基本氨基酸? 为了回答这一问题，我们需要考虑：有多少个 2-元素 DNA 序列? 还是用树图形式表示这一问题的答案，如图 2.2 所示. 由图可知，存在 16 种可能的 2-元素 DNA 序列. 第一个元素有四种选择，而对应于其中每一种选择，第二个元素有四种选择；读者将注意到所有可能的 2-元素 DNA 序列为 4×4=16 种. 注意，2-元素序列不足以编码所有 20 种不同的基本氨基酸. 事实上，3-元素序列才足以编码 20 种不同的基本氨基酸. 一个简单的计数过程就已证明为什么至少需要 3 个元素. ■

⊖ 本章是由 Helen Marcus-Roberts，Fred S. Roberts 和 Barry A. Tesman 编写的.

上面给出的两个例子说明了下面的基本规则.

乘法规则　如果某事件能以 n_1 种方式发生，且不管第一个事件是如何发生的，第二个事件都能以 n_2 种方式发生，那么这两个事件合起来能以 $n_1 \times n_2$ 种方式发生. 更一般地，如果某事件能以 n_1 种方式发生，且无论第一个事件是如何发生的，第二个事件都能以 n_2 种方式发生，而且无论前两个事件是如何发生的，第三个事件能以 n_3 种方式发生，以此类推，那么所有事件合起来能以

$$n_1 \times n_2 \times n_3 \times \cdots$$

种方式发生.

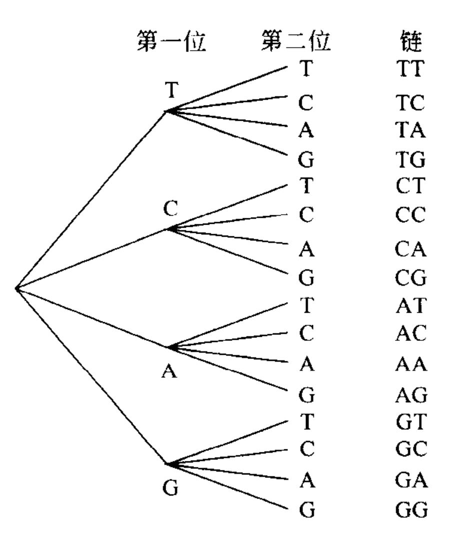


图 2.2　计数 2-元素 DNA 序列数量的树图

再回到位串的例子，通过乘法规则我们立刻看到长度正好等于 3 的位串的数量是 $2\times2\times2=2^3=8$，因为第一位(0 或 1)有两种选择，且无论怎么选择第二位有两种选择，且无论前两位是如何选择的，第三位都有两种选择. 类似地，在例 1.3 的管道网络问题中，如果 3 个连接点处的每一处的管道尺寸有 7 种选择，那么有

$$7\times7\times7=7^3=343$$

种不同的可能网络. 如果有 40 个连接点，那么有

$$7\times7\times\cdots\times7=7^{40}$$

种不同的可能网络. 注意，根据我们在第 1 章中的观察，通过枚举出(列出)所有可能的管道网络来计数它们的数量是不可行的，需要枚举法之外的其他计数方法. 乘法规则给出了这样的一个方法. 在本书的较前部分，我们将研究这样的简单计数方法.

接下来，假设 A 是有 a 个对象的集合，B 是有 b 个对象的集合. 那么从 A 中挑选出一个对象，然后再从 B 中挑选出一个对象的方法的数量是 $a\times b$. 这是乘法规则的一个更精确描述.

作为最后的例子，3-元素 DNA 序列的数量是

$$4\times4\times4=4^3=64.$$

这就是不同的 3-元素序列足以编码所有 20 种不同的基本氨基酸的原因；事实上，几种不同的链编码同一个氨基酸，这不同于莫尔斯码的情况，在莫尔斯码中，编码字母表中所有 26 个字母需要至多 4 位的位串，但并非每一种可能的位串都要使用. 在 2.9 节中，我们将考虑 Gamow[1954a, b] 提出的关于哪些 3-元素序列可编码相同氨基酸的提议.

继续以 DNA 分子为例，我们看到 4 个基的序列的数量是 4^4，当基为 100 时这个数量为 4^{100}. 完整的 DNA 分子有多长呢? 表 2.1 给出部分答案. 注意：在黏液菌中，基因组有 5.4×10^7 个基或基对. 因此，这样的序列数量为

$$4^{5.4\times10^7},$$

这个数字大于

$$10^{3.2\times10^7}.$$

$10^{3.2\times10^7}$ 是 1 后面跟着 3.2×10^7 个 0! 这个数字大得无法想象. 类似的结果对其他的有机体也同样成立. 通过对所有可能性的简单计数，我们就能理解在遗传的组成过程中的巨大的可能性变化. 给定可能的 DNA 序列的数量，在自然界中存在如此惊人的变化，那么两个个体不可能相同就根本没有什么可惊讶的了. 我们已不只一次提到，给定如此巨大的可能性数量，通过简单的枚举这些可能性的方法是不可能计数它们的，有必要开发计数规则或过程，这些规则或过程不通

过罗列它们就可以计数可能性的数量．这是组合数学的三个基本问题中的一个：开发不通过枚举进行计数的过程．

表 2.1　各种有机体的可能 DNA 序列的数量

门	属和物种	基因组大小(基对)	可能序列的数量
藻类	盐藻	6.6×10^5	$4^{6.6\times10^5}>10^{3.97\times10^5}$
支原体	肺炎支原体	1.0×10^6	$4^{1.0\times10^6}>10^{6.02\times10^5}$
细菌	大肠杆菌	4.2×10^6	$4^{4.2\times10^6}>10^{2.52\times10^6}$
酵母菌	啤酒酵母	1.3×10^7	$4^{1.3\times10^7}>10^{7.82\times10^6}$
黏液菌	黏菌	5.4×10^7	$4^{5.4\times10^7}>10^{3.25\times10^7}$
线虫	线虫	8.0×10^7	$4^{8.0\times10^7}>10^{4.81\times10^7}$
昆虫	黑腹果蝇	1.4×10^8	$4^{1.4\times10^8}>10^{8.42\times10^7}$
鸟	家鸡	1.2×10^9	$4^{1.2\times10^9}>10^{7.22\times10^8}$
两栖动物	爪蟾	3.1×10^9	$4^{3.1\times10^9}>10^{1.86\times10^9}$
哺乳动物	晚期智人	3.3×10^9	$4^{3.3\times10^9}>10^{1.98\times10^9}$

资料来源：Lewin[2000].

尽管 DNA 序列的数量巨大，但是“排序”或“绘制”不同有机体(包括人类基因组)已变得可行，这部分归功于组合数学方法的使用．基因(gene)是携带制造特殊蛋白质代码的一条 DNA．“绘制”基因组将需要定位它的每一个基因；“排序”基因组将需要确定形成每个基因的基的精确顺序．人类的基因组包含大约 100 000 个基因，而且每一个基因都有上千或更多的基．在绘制或排序基因组中更多的关于组合数学的使用可以参见 Clote 和 Backofen[2000]、Congress of the United States[1988]、Farach-Colton 等[1999]、Gusfield[1997]、Lander 和 Waterman[1995]、Pevzner[2000]、Setubal 和 Meidanis[1997]或 Waterman[1995]．　18

【例 2.3 电话号码】 有一个地区电话号码曾经是由两个字母后面跟着五个数字组成的．存在多少不同的电话号码呢？使用乘法规则，有人给出这样的答案：

$$26\times26\times10\times10\times10\times10\times10=26^2\times10^5.$$

尽管这一计数是正确的，但它不是一个好答案，因为电话号码盘上相同位置的两个字母将导致相同的电话号码．读者也许希望想象一个电话号码盘．(电话号码盘示意图如图 2.3 所示．)所有数字键上都有三个字母，1 和 0 除外．因此，字母 A，B，C 是等效的，W，X，Y 也一样，以此类推．事实上，只存在 8 种不同的字母．因此，不同的电话号码数量是

$$8^2\times10^5=6.4\times10^6.$$

所以，有略多于 6 百万个这样的号码．在 20 世纪 50 年代和 20 世纪 60 年代，大多数地区的号码都变成简单的七位数字号码，且前两个数字限制为既不能是 0 也不能是 1 组成的．但电话号码的数量仍然是 $8^2\times10^5$．长途直拨号码是通过加上三个数字区域代码实现的．区域代码不能开始于 0 或 1，且区域代码中间必须有 0 或 1．使用这些限制，我们计算得到可能的电话号码数量是

$$8\times2\times10\times8^2\times10^5=1.024\times10^9.$$

这足以服务于超过 10 亿名顾客．为了服务于更多的顾客，长途直拨号码变成以一个 1 开始的 11 位数字作为长途呼叫，这解除了区域代码在中间必须有一个 0 或一个 1 的限制．电话号码的数量变成

$$1\times8\times10\times10\times8^2\times10^5=5.12\times10^9.$$

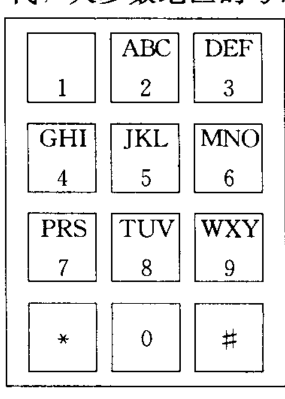


图 2.3　电话号码盘　19

随着技术的不断改进，电信业能够拥有以 1 开始且对后面的 10 个数字没有限制的所有号码. 因此，现在有 10^{10} 个可能的电话号码. 然而，对号码数量的需求继续非常快地增加(例如，多线、传真机、手机、寻呼机，等等). 当 10^{10} 也不够用的时候，我们将做什么呢? ■

【例 2.4 开关函数】 设 B_n 是所有长度为 n 的位串的集合. n 变量开关函数(switching function，即布尔函数(Boolean function))是给每一个长度为 n 的位串指定一个数(0 或 1)的函数. 例如，设 $n=2$，那么 $B_2=\{00, 01, 10, 11\}$. 在 B_2 上定义的两个开关函数 S 和 T 如表 2.2 所示. 数字计算机的细节设计问题通常涉及寻找实现特定函数行为的实际电路. 一台计算机设备实现两变量、三变量或四变量的开关函数. 现在，通过互连电子网络可以用多种方法实现每一个开关函数. 计算机工程师不是从头开始解决给定开关函数的高效设计问题，而是希望对应于每一个开关函数都有一个高效网络实现的目录. 遗憾的是，这乍看起来似乎是一个不可能实现的目标. 有多少个 n 变量开关函数呢? 通过扩展例 2.1，集合 B_n 中有 2^n 个元素. 因此，根据乘法规则，存在 $2\times2\times\cdots\times2$ 个不同的 n 变量开关函数，在这个乘积中有 2^n 个项，即总共有 2^{2^n} 个不同的 n 变量开关函数. 即使 $n=4$，这样的函数也有65 536个，而且这个数字增加得非常快. 幸运的是，通过利用对称性，就其涉及的计算而言可以认为某些开关函数是等价的. 因此，我们无需为每一个开关函数确定最佳设计，而只需要为一部分开关函数确定最佳设计，而其他开关函数等价于我们设计的开关函数. 在构建第一批计算机时，哈佛大学的研究员小组花费了很大功夫枚举出了所有 4 变量的开关函数，并确定出其中哪些是等价的. 他们发现可以把每一个开关函数归约到 222 种类型之一中(Harvard Computation Laboratory Staff[1951]). 在第 8 章，我们将说明如何从波利亚(George Pólya)的一个强有力的定理得到诸如此类的结果. 关于开关函数的更详细讨论，参见 Deo[1974, Ch. 12]、Harrison[1965]、Hill 和 Peterson[1968]、Kohavi[1970]、Liu[1977]、Muroga[1990]、Pattavina[1998]、Prather[1976]、Stone[1973]. ■

20

表 2.2　两开关函数

位串 x	$S(x)$	$T(x)$
00	1	0
01	0	0
10	0	1
11	1	1

【例 2.5 食物过敏】 一名过敏症专科医生给一位饭后经常严重腹痛的病人看病. 他怀疑某些特定的食物有问题：土豆、巧克力、玉米和花生. 此时不十分清楚的是这一病症是因为其中的一种食物，还是它们相互作用后所引发的. 这名医生告诉病人尝试这些食物的不同组合，看一看是否存在食物间的相互作用. 那么必须尝试多少种不同的组合呢? 每一种食物可以出现于组合中，也可以不出现. 因此，有 $2\times2\times2\times2=2^4=16$ 种可能的组合. 理论上，基于这四种食物有 2^{2^4} 种可能的食物过敏表现；食物的每一种可能组合或者带来过敏反应或者不带来过敏反应. 每一个人对这些食物组合的个体敏感度对应于一个开关函数 $S(x_1, x_2, x_3, x_4)$，其中，如果在食物中有土豆那么 x_1 等于 1，如果在食物中有巧克力那么 x_2 等于 1，如果在食物中有玉米那么 x_3 等于 1，如果在食物中有花生那么 x_4 等于 1. 例如，食物中含有土豆或含有玉米和花生时有过敏反应的人应该明确地显示有开关函数 S，且这个开关函数有 $S(1, 0, 0, 0)=1$，$S(1, 1, 0, 0)=1$，$S(0, 0, 1, 1)=1$，$S(0, 1, 1, 0)=0$，等等. 实际上，如果变量的数量很大，要想知道一个开关函数所有可能的位串的值是不可能的；存在太多的可能位串. 那么，这一实际问题是要开发猜测部分定义的开关函数的值的方法. 关于这一问题最近有很多研究. 例如，参见 Boros 等[1995]、Boros，Ibaraki 和 Makino[1998]、Crama，Hammer 和 Ibaraki[1988]、Ekin，Hammer 和 Kogan[2000]. 当某个器件失效时，类似的因果问题出现于拥有失效记录的复杂电子系统的故障诊断中(在例 2.21 中我们对此给出更详细的说明). 同样，这类因果问题也出现于教授机器人在带有某些障碍的区域操作中，在这个区域中，障碍可能以特定的暗或亮的像素模式出现，在某

些情形中，像素的模式对应于一个对象，而在另外一些情形中，像素的模式不对应于对象. 对于其他应用，参见 Boros 等[2000]. ■

本节练习⊖

1. 宾夕法尼亚州卡莱尔的人口大约是 20 000. 如果每一位居民的名字有三个字头，那么至少有两名居民的名字的字头是相同的，这是真的吗？请解释你的答案. [21]
2. 一个图书馆有 5000 本书，图书管理员想要使用包含 3 个字母其后跟着 3 个数字的代码字编码每一本书. 在使用不同的代码字的情况下，有足够多的代码字编码所有 5000 本书吗？
3. (a)继续使用第 1 章的练习 7，计算长度至多为 3 的三进制代码串的最大数量.
   (b)对于长度至多为 4，重复上述的问题.
   (c)对于长度正好等于 4，但开始于 0 或 1 的三进制代码，重复上述的问题.
4. 在我们关于电话号码的讨论中，假设我们保持例 2.3 中对区域代码的最初限制，要加长这一地区的电话号码，允许它是任意的 8 位数字号码且前三个数字中的任意一个都不是 0 或 1. 存在多少个地区电话号码？包含区域代码在内的地区电话号码有多少？
5. 如果我们想要使用长度至多等于 n 的位串来编码字母表中的所有 26 字母及所有 10 个十进制数字，那么最小的 n 是多少？(对于莫尔斯码来说，n 是多少?)
6. 每一项是 0 或是 1 的 $m\times n$ 矩阵有多少个？
7. 一个乐队必须至少有一个成员. 它可以有至多一位鼓手，至多一位钢琴手，至多一位低音提琴手，至多一位领唱和至多两位伴唱. 如果我们认为任意两名鼓手没有差异，且对其他种类的乐手也同样成立，因此当两个乐队中每一种类的乐手的人数相同时称这两个乐队相同. 这时，有多少种可能的乐队？验证你的答案.
8. 有多少小于 100 万的非负整数包含数字 2?
9. 枚举所有两变量开关函数.
10. 如果一个函数把 0 或 1 指定给每一个 n 变量的开关函数，那么有多少种这样的函数？
11. 一个开关函数称为是*自对偶的*(self-dual)，如果当交换位串中的 0 和 1 时位串的值 S 不变. 例如，表 2.2 中的函数 S 是自对偶的，但是其中的函数 T 不是自对偶的. n 变量的自对偶函数有多少种？
12. (Stanat 和 McAllister[1977])某些计算机使用长度为 p 的位串表示一个整数(正或负的). 位串中的最后一位表示符号，前 $p-1$ 位被用于编码这一整数. 对于给定的 p，能以这种方式表示的不同整数的最大数量是多少？如果 0 必须是其中的一个整数，情况又如何(0 的符号是+或−)?
13. (Stanat 和 McAllister[1977])每一个整数都可以(不唯一地)用 $a\times 2^b$ 表示，其中 a，b 是整数. 整数的*浮点表示*(floating-point representation)使用一个长度为 p 的位串来表示一个整数，其方法是使用前 m 个位编码 a，且使用其余的 $p-m$ 个位编码 b，后两位的编码按练习 12 中所描述的方式进行.
    (a)对于给定的 p，使用浮点表示的不同整数的最大数量是多少？
    (b)如果浮点表示是按下面的方式进行的：编码数字 a 的前导位是 1，重复(a)的问题.
    (c)如果必须包含 0，重复(a)的问题. [22]
14. 在进行借贷申请时，银行需要根据历史记录得出结论：借贷申请人拥有某些综合特征才有希望能够偿还他的借贷，而拥有其他综合特征的人就无法偿还他的借贷. 作为他们的主要特征，假设一家银行使用：
    婚姻状况：已经结婚，单身(从没有结过婚)，单身(以前曾经结过婚).
    过去的借贷：以前有拖欠，以前没有拖欠.

⊖ 对于第 2 章的练习，每一节后面的练习可能使用前面某一节(必修)的技术，而未必一定要使用刚刚学到的技术. 还有，在本章的末尾有附加练习. 实际上，有些练习不使用当前节的技术. 为了理解一个新技术，你必须理解何时使用它及何时不使用它.

职业：受雇用，失业(在一年内)，失业(超过一年).

(a)当考虑这些特征时，有多少种不同的借贷申请情况？

(b)当考虑这些特征时，有多少种可能的借贷偿还/拖欠行为？

2.2　加法规则

现在我们讲述第二个基本计数规则. 考虑下面的例子.

【例 2.6 国会代表团】 有 100 名参议员和 435 名众议院成员. 要挑选一个代表团会见总统. 如果这个代表团是由一名参议员*和*一名众议员组成的，那么有多少种不同的方法可以选出这样的代表团？根据乘法规则，答案是

$$100\times 435=43\,500.$$

如果这个代表团是由一名参议员*或*一名众议员组成的，那么答案又如何呢？于是存在

$$100+435=535$$

种可能的代表团. 这一计算说明了计数的第二个基本规则，加法规则. ■

加法规则　*如果一个事件以 n_1 种方式发生且第二个事件以 n_2 种(不同)方式发生，那么存在 n_1+n_2 种方式，其中或者第一个事件发生或者第二个事件发生(但不是二者同时发生). 更一般地，如果第一个事件以 n_1 种方式发生，第二个事件以 n_2 种(不同)方式发生，第三个事件以 n_3 种(仍然不同)的方式发生，……，那么存在*

$$n_1+n_2+n_3+\cdots$$

种方式使其中(正好)一个事件发生.

在例 2.6 中，我们用楷体字表示词"和"与"或". 这些关键字通常说明加法规则是合适的，还是乘法规则是合适的. 词"和"表明乘法规则合适；词"或"表明加法规则合适.

[23]

【例 2.7 草案筛选】 一个职业足球队需要制定两个选择草案，且对其选择的限制是 3 名四分卫，4 名后卫和 5 名边接应队员. 为了挑出一名四分卫和一名后卫，根据乘法规则有 $3\times 4=12$ 种方法. 如果两名选手必须踢不同的位置，那么挑选出两名选手有多种少方法？你可以挑选或者一名四分卫和一名后卫、一名四分卫和一名边接应队员或者是一名后卫和一名边接应队员. 根据前面的计算，完成第一种挑选有 12 种方法，完成第二种挑选有 15 种方法(为什么)，完成第三种挑选有 20 种方法(为什么). 因此，根据加法规则，从不同的位置挑选两名选手的方法共有

$$12+15+20=47$$

种. ■

【例 2.8 BASIC 和 JAVA 中的变量】 程序设计语言 BASIC(全称是初学者通用符号指令代码)要追溯到 1964 年. BASIC 的早期实现中的变量名是一个字母，一个字母后面跟着一个字母或者一个字母后面跟着一个十进制数字，即数字 0，1，2，3，4，5，6，7，8，9 中的一个. 有多少个可能的变量名呢？根据乘法规则，对后面两种情况的变量名分别有 $26\times 26=676$ 和 $26\times 10=260$ 个. 根据加法规则，总共存在 $26+676+260=962$ 个变量名.

对更多变量名的需求是发展更高级程序设计语言的原因之一. 例如，1995 年引入的 JAVA 程序设计语言的变量名长度从 1 到 65 535 个字符. 每一个字符可以是一个字母(大写字母或小写字母)、一个下划线、一个美元符号或一个十进制数字，只是第一个字符不能是十进制数. 通过使用加法规则，我们看到可能的字符数量是 $26+26+1+1+10=64$，因为第一个字符不能是十进制数，所以有 $64-10=54$ 种可能. 最后，通过使用加法和乘法规则，我们看到变量名的数量是

$$54\cdot 64^{65\,534}+54\cdot 64^{65\,533}+\cdots+54\cdot 64+54.$$

这一定能满足更多变量的需求.㊀ ■

在这一节结束之际，我们以下面的方式重述加法规则. 假设 A 和 B 是不相交的集合，且我们希望刚好从 A 或从 B 中挑选出一个元素. 那么挑选这个元素的方法总数是 A 中元素的数量加上 B 中元素的数量.

本节练习

1. 长度为 3，4 或 5 的位串有多少个?
2. 要从 8 名科学家、7 名物理学家和 12 名牧师中选定一个委员会. 如果这个委员会需要有两名不同背景的成员，那么这样的委员会有多少个? [24]
3. 考虑五位数，它的每一位数字都是{1，2，…，9}中的一个，且所有数字或者都是偶数或者都是奇数. 这样的数有多少个?
4. 每一张美国运通卡有一个 15 位数字的数，计算机使用这个数来识别该卡. 如果每一个数字可以是 0 到 9 之间的任意一个数，那么对于 1 千万名信用卡持有者来说，有足够多的不同账号吗? 如果这些数字只能是 0 或 1，情况又如何?
5. 有多少个或者开始于 d 或者没有字母 d 的 5 字母字?
6. 当投掷两枚骰子时，我们有多少方法可以得到和为 3 或和为 4 的点数?
7. 假设管道网络有 30 个连接点，对于每一个连接点，有 2 个选择：管道可以是 7 个尺寸中的一个及 3 种材料中的一种. 有多少种不同的管道网络?
8. 长度为 3 且或者没有 C 或者在第一个位置没有 T 的 DNA 链有多少种?

2.3 排列

在组合数学中，我们经常谈论 n 元素集合，即由 n 个不同元素组成的集合. 为了方便，我们称它们为 n 集合(n-set). 一个 n 集合的排列(permutation)是这个集合中元素的按序排列. 计算 n 集合的排列数量通常很重要.

【例 2.9 工作面试】 Jordan 女士、Harper 先生和 Gabler 女士三人参加工作面试. 他们一共有多少种不同的接受面试的顺序? 我们可以如下列出所有可能的面试顺序：

1. Jordan, Harper, Gabler
2. Jordan, Gabler, Harper
3. Harper, Jordan, Gabler
4. Harper, Gabler, Jordan
5. Gabler, Jordan, Harper
6. Gabler, Harper, Jordan

我们看到有 6 种可能的顺序. 另外，可以发现第一个接受面试的人有三种选择. 对于其中的每一种选择，第二个接受面试的人有两种选择. 对其中每一种选择，第三个接受面试的人只有一种选择. 因此，根据乘法规则，接受面试的可能顺序数量是

$$3\times2\times1=6.$$

每一个顺序就是一个排列. 我们要求由 Jordan、Harper、Gabler 组成的 3 集合的排列数量.

如果有 5 个人接受面试，那么接受面试的可能顺序的计数仍然可以通过枚举法获得；然而，那样做相当乏味. 不难发现第一个接受面试的人有 5 种可能，而第二个接受面试的人有 4 种可

㊀ 这个和的第一项 $54\cdot64^{65\,534}$ 的值大约是 $8.527\times10^{118\,367}$. 可以证明，宇宙中原子粒子的数量级至少是 10^{80}(参见 Dembski[1998]).

能，以此类推，于是产生下面的结果：

[25]

$$5\times4\times3\times2\times1=120.$$

这是所有的接受面试人的可能顺序. ■

扩展例 2.9 的计算给出如下结果：n 集合的排列数量可以由下式给出

$$n\times(n-1)\times(n-2)\times\cdots\times1=n!.$$

在例 1.1 中，我们讨论了取五种不同药物的顺序的数量. 这相当于 5 集合的排列数量，所以它是 $5!=120$. 为了再一次弄清为什么通过枚举法不可能计算出其数量，我们在表 2.3 中给出几个 $n!$ 的值.

给出一个例子，数 25! 就已经是一个相当大的数了，大到了难以想象的程度. 为了认识到这一点，注意

$$25!\approx1.55\times10^{25}.$$

一台每秒可以检查 10 亿个排列的计算机也几乎需要 5 亿年来检查 1.55×10^{25} 个排列.[⊖] 尽管有上面的结果，但是仍然有很多情况，利用枚举法给出 n 集合的所有的排列数量是很有用的. 在 2.16 节中，我们将给出枚举所有可能排列的算法.

数 $n!$ 可以用 $s_n=\sqrt{2\pi n}(n/e)^n$ 的计算来近似. 使用 s_n 近似 $n!$ 称为斯特林近似(Stirling's approximation). 为了弄清这一近似是如何好，注意它把 5! 近似为 $s_5=118.02$ 而 10! 近似为 $s_{10}=3\,598\,600$. (与表 2.3 中的真实值相比较.)通过当 n 趋近∞时 $n!$ 与 s_n 的比率趋近于 1 这一事实可以证明这一近似的质量(另一方面，当 n 趋近于∞时差 $n!-s_n$ 趋近于∞). 对于证明，可以参见高级微积分课本，如 Buck[1965].

表 2.3 n 从 0 到 10 时 $n!$ 的值

n	0	1	2	3	4	5	6	7	8	9	10
$n!$	1	1	2	6	24	120	720	5040	40 320	362 880	3 628 800

本节练习

1. 列出下面集合的所有排列.
   (a){1, 2, 3}　　(b){1, 2, 3, 4}

[26]

2. {1, 2, 3, 4, 5}的开始于 5 的排列有多少？
3. {1, 2, 3, …, n}的开始于 1 且结束于 n 的排列有多少？
4. 计算下列 n 的 s_n 且把它与 $n!$ 比较.
   (a)$n=4$　　(b)$n=6$　　(c)$n=8$
5. {1, 2, 3, 4}的开始于奇数的排列有多少？
6. (a){1, 2, 3, 4, 5}的排列中第二个位置为 2 的排列有多少？
   (b){1, 2, 3, …, n}($n\geq3$)的排列中第二个位置为 2 且第三个位置为 3 的排列有多少？
7. 有五名身高不同的篮球新队员. 如果按照最高的人一定在第一位，最矮的人一定在最后一位的顺序排列，那么有多少种方法排列这五个人？
8. (Cohen[1978])
   (a)在 6-汽缸发动机中，偶数汽缸在左边而奇数汽缸在右边. 一个好的点火顺序是数 1 到 6 的一个排列，

⊖ 为了弄清其原因，注意 1 年大约有 3.15×10^7 秒. 因此，每秒可以检查 10 亿个排列的计算机可以在 1 年内检查 $3.15\times10^7\times10^9=3.15\times10^{16}$ 个排列. 因此，检查 1.55×10^{25} 个排列所需要的年数是

$$\frac{1.55\times10^{25}}{3.15\times10^{16}}\approx4.9\times10^8.$$

且在这一排列中左右两侧是相互交替的. 开始于左边汽缸的可能的好的点火顺序有多少种?

(b)对于 $2n$-汽缸发动机, 上述的问题又如何?

9. 10 名工作申请人被邀请进行面试, 有五个人被告知上午参加面试, 有五个人被告知下午参加面试. 安排面试的不同顺序有多少种? 如果所有十个申请人都被安排在上午面试, 有多少不同顺序安排面试? 比较这两个答案.

2.4 计算的复杂度

我们已经发现并不是所有的组合数学问题都可以在计算机上解决, 至少不能通过枚举法得以解决. 假设一个计算机程序实现解决一个组合数学问题的算法. 在运行这样的程序之前, 我们希望知道这个程序是否能在"合理的"时间总量内运行, 且所使用的存储量或内存量不超过"合理的"(或允许的)存储量或内存量. 一个程序所需要的时间或存储依赖于输入. 为了衡量一个程序的运行成本, 我们尝试计算一个*成本函数*(cost function)或*复杂度函数*(complexity function). 这是一个根据所需要的时间或所需要的存储量衡量成本的函数 f, 是作为输入问题尺度 n 的函数. 例如, 我们也许要问处理两个 n 行、n 列方阵的乘积需要多少个操作. 这个操作数量就是 $f(n)$.

通常, 在一台特定的机器上运行一个特定的计算机程序的成本将随着程序员的技巧和机器的特性而发生变化. 因此, 现代计算机科学更加强调算法的比较而不是程序的比较, 强调算法的复杂度 $f(n)$ 的估算, 估算与所实现算法的特定程序及机器无关. 计算算法复杂度的愿望是组合数学技术发展的一个主要推动力.

27

【例 2.10 卖货郎问题】 一名卖货郎希望遍访 n 个不同的城市, 在第一座城市开始并结束他的工作旅程. 他不关心到访城市的顺序, 关心的是最小化旅程的总成本. 假设从城市 i 到城市 j 的旅程成本为 c_{ij}. 问题是要寻找一个计算最便宜路程的算法, 其中一条路线的成本是该路线中的连结成本 c_{ij} 的和. 这是一个典型的组合优化问题.

对于卖货郎问题, 我们将牵涉到枚举算法: 枚举所有可能的路线并计算每一条路线的成本. 我们将尝试计算这一算法的复杂度, 其中 n 是输入的大小, 即城市的数量. 假设对每一条路线, 识别该路线及计算其成本都是可比较的, 且要花费一个时间单位.

现在, 开始且结束于城市 1 的任意路线对应于余下的 $n-1$ 个城市的一个排列. 因此, 存在 $(n-1)!$ 条这样的路线, 所以 $f(n)=(n-1)!$ 个时间单位. 我们已证明这个数字非常大. 当 $n=26$, $n-1=25$ 时, 我们已证明 $f(n)$ 非常大以至于通过计算机执行这一算法都是不可能的. 我们将在 11.5 节再次回到卖货郎问题. ■

有趣的是, 很多猜测问题中都出现卖货郎问题. 例 2.11 到例 2.16 给出实际应用中这一问题的若干变形.

【例 2.11 自动取款机(ATM)问题】 银行有很多 ATM. 每一天, 信使都会一台机器一台机器地收集汇总计算机信息等. 为了最小化所花费的时间, 以什么样的顺序遍历这些机器呢? 在实际中很多银行都会发生这一问题. ATM 的初期, 第一批使用卖货郎算法解决这一问题的一家银行是波士顿的 Shawmut Bank[⊖]. ■

【例 2.12 电话亭问题】 每周一次, 要遍历一个地区的每一个电话亭, 收集硬币. 为了最小化所花费的时间, 应该以什么样的顺序遍历呢? ■

⊖ 这个例子来自于 Margaret Cozzens(私人通信).

【例 2.13 自动货栈中的机器人问题】 将来的自动货栈将由机器人来取货. 想象一个药物货栈，其中堆满按行和列排列的货架. 有一份订货单需要 10 箱阿斯匹林、6 箱清洗剂、8 箱邦迪等等. 每一种药物都按照行、列和高度定位. 为了最小化所需要的时间，机器人按什么样的顺序取货? 我们需要程序化这个机器人以便解决这一卖货郎问题. (参见 Elsayed[1981]、Elsayed 和 Stern[1983].) ■

28

【例 2.14 X 射线结晶学问题】 在 X 射线结晶学中，我们必须以一系列指定的角度移动衍射仪. 一个接一个这样做需要以时间和设置为成本. 我们如何才能最小化这一成本呢? (参见 Bland 和 Shallcross[1989].) ■

【例 2.15 制造业】 很多工厂必须做很多工作或必须运行很多工序. 运行工序 i 之后，在我们运行工序 j 之前可能存在某种设置成本；即为下一工序做准备所需要的时间或金钱或劳动力方面的成本. 有时候，这一成本很小(例如，简单地做一些小的调整)，而有时候它又比较显著(例如，设备的完全清理或新设备的安装). 为了最小化总成本，各工序应以什么样的顺序运行? (对此的更多运用，参见例 11.5 和 11.6.3 节.) ■

【例 2.16 电路板中的孔】 在 1993 年，Applegate、Bixby、Chvátal 和 Cook(参见http://www.cs.rutgers.edu/~chvatal/pcb3038.html)发现了当时得以解决的最大 TSPLIB[㊀] 卖货郎问题的解. 该问题有 3038 个城市，来自于实际中为了制造一块有 3038 个孔的电路板所涉及的最高效的钻孔顺序问题(这是卖货郎问题的另外一个应用). 对此的更多信息，可参见 Zimmer[1993].[㊁] ■

卖货郎问题是激发研究人员赋出努力寻找"好"算法的一个例子. 的确，这一问题属于著名的 *NP 完全问题*(NP-complete)或 NP *难题*(NP-hard problem)，对于这一类问题，严格来讲不存在好的算法. 我们将在 2.18 节讨论该问题，在那里我们将简要地定义 NP 完全问题，而且我们定义一个算法是"*好的*"，如果这一算法的复杂度函数 $f(n)$ 被 n 的多项式界定. 这样的算法称为*多项式算法*(polynomial algorithm)，更精确地，称为多项式时间算法.

【例 2.17 调度计算机系统[㊂]】 一个计算机中心需要运行 n 个程序. 每一个程序需要一定的资源，例如编译器、一组处理器且每一个处理器需要一定量的内存. 我们通常把所需的资源称为对应于这一程序的*配置*(configuration). 系统从第 i 个配置到第 j 个配置的转换会发生与其相关的成本 c_{ij}. 例如，如果两个程序需要类似的配置，那么相继运行它们是有利的. 计算机中心希望最小化对应于运行的 n 个程序的总成本. 运行每个程序的固定成本不随程序的运行顺序的不同而发生变化. 唯一发生变化的是转换成本 c_{ij}. 因此，计算机中心希望寻找一种运行程序的顺序，使得总的转换成本达到最小. 在运筹学中有很多类似的调度问题. 我们将在 11.5 节和 11.6 节进一步讨论它们. 与卖货郎问题一样，枚举程序运行的所有可能顺序的算法是不可行的，因为显然它有一个 $n!$ 的计算复杂度. (为什么是 $n!$ 而不是 $(n-1)!$ 呢?)的确，从形式的角度看，这一问题和卖货郎问题几乎是等价的，只需简单地把城市替换成配置就可以了. 求解其中的一个问题的

29

㊀ TSPLIB(http://www.iwr.uni-heidelberg.de/iwr/comopt/software/TSPLIB95/)是卖货郎问题 110 个实例的数据库.

㊁ 到 1998 年，Applegate、Bixby、Chvátal 和 Cook 已经解决了更大的 TSPLIB 实例：例如，关于人口不少于 500 人的 13 509 个美国城市的最优旅行问题. (参见 http://www.cs.rutgers.edu/~chvatal/usa13509.html.)本书完成之际，TSPLIB 中还有四个问题没有解决，其中有 14 051、18 512、33 810 和 85 900 个城市. 对于卖货郎问题的计算方面的研究，可以参见 Applegate 等[1998]. 也可以参见卖货郎问题的主页(http://www.math.princeton.edu/tsp/index.html).

㊂ 这个例子得自于 Stanat 和 McAllister[1977].

任意算法很容易转换成求解另一问题的算法. 这是使用数学技术解决实际问题的一个主要动机，即我们可以先解决一个问题，于是很快就会有适用于其他大部分问题的技术，而这些问题在表面上看完全不同. ■

【例 2.18 在文件中搜索】 在确定计算复杂度的过程中，我们不是总能精确地知道计算要花多长时间. 例如，考虑搜索 n 个键(识别数字)的列表来寻找某个人的键以便存取该人的文件. 现在，有可能出现的情况是该人的键可能处于列表中的第一个位置. 然而最坏的情况是，这个键是列表中的最后一个. 处理最坏可能性的成本有时用来度量计算复杂度，称为*最坏情况复杂度*(wrost-case complexity). 这里，$f(n)$应该与 n 成正比. 另一方面，计算复杂度的另一个合适的度量是处理一种情况的*平均*(average)成本，即*平均情况复杂度*(average-case complexity). 假设所有情况都是平等的，那么计算处理每一种情况的成本，再把这些成本加起来，然后除以情况总数就可以计算得到平均成本. 在我们的例子中，平均情况复杂度与$(n+1)/2$成正比，假设作为搜索对象所有键是平等的，对于处理 n 个情况的成本的和为 $1+2+3+\cdots+n$. 因此，使用这一和的标准公式我们可以得到

$$f(n)=\frac{1}{n}(1+2+\cdots+n)=\frac{1}{n}\frac{n(n+1)}{2}=\frac{n+1}{2}.$$ ■

在 3.6 节，我们将讨论存储文件的二叉搜索树的使用，并证明通过使用二叉搜索树可以大幅度减小寻找给定键的文件的计算复杂度.

本节练习

1. 如果一台计算机一秒钟可以处理 10 亿个顺序，如果 $n=25$，那么通过枚举法它需要花多少年解决例 2.17 的计算配置问题?
2. 如果一台计算机一秒钟可以处理 1000 亿个顺序而不是 10 亿个顺序，如果 $n=26$，那么通过枚举法它要花多少年解决卖货郎问题?(计算机速度上的改进能给 18 页上的基于脚注㊀的结论带来很大的不同吗?) [30]
3. 考虑 n 个立法委员会有序参加 n 个连续时间间隔的会议的调度问题. 每一个委员会主席都声明哪一个时间间隔是他的首选，我们设法安排这些会议使得得到首选会议时间的委员会主席的数量尽可能多. 假设我们通过枚举所有可能的安排来解决这一问题，且对于每一个安排我们将计算得到首选会议时间的委员会主席的数量. 这一过程的计算复杂度是什么?(对计算得到首选的委员会主席数量的步骤做一个假设.)
4. 假设在一个地区有 n 个电话亭，且我们希望遍历它们. 每个电话亭访问两次，但不能连续访问. 讨论寻找最小化总遍历时间的遍历顺序的朴素算法的计算复杂度.
5. 通过枚举法解决卖货郎问题. 其中 $n=4$，而且成本 c_{ij} 是由下面的方阵给出的：

$$(c_{ij})=\begin{bmatrix} - & 1 & 8 & 11 \\ 16 & - & 3 & 6 \\ 4 & 9 & - & 11 \\ 8 & 3 & 2 & - \end{bmatrix}.$$

6. 解决例 2.17 的计算机系统调度问题. 其中 $n=3$，且从第 i 个配置到第 j 个配置的转换成本由下面的方阵给出：

$$(c_{ij})=\begin{bmatrix} - & 8 & 11 \\ 12 & - & 4 \\ 3 & 6 & - \end{bmatrix}.$$

7. 假设检查一个列表中的每一个键所花的时间为 3×10^{-9}. 如果存在 n 个键，且我们依次搜索它们直到发现正确的键，求：

   (a)最坏情况复杂度　　(b)平均情况复杂度

8. 假设检查每一个键所花的时间为 3×10^{-11}，重复练习 7.
9. (Hopcroft[1981])假设 L 是长度为 n 的位串的一个集合. 假设 A 是确定给定长度为 n 的位串是否在 L 中的算法. 假设 A 总是需要花 2^n 秒给出答案. 那么 A 有相等的最坏情况复杂度和平均情况复杂度 2^n. 假设 $\hat{L}$ 是由形如

$$x_1x_2\cdots x_nx_1x_2\cdots x_n$$

的所有位串组成的，其中 $x_1x_2\cdots x_n$ 在 L 中. 例如，如果 $L=\{00, 10\}$，那么 $\hat{L}=\{0000, 1010\}$. 考虑下面确定给定长度为 $2n$ 的位串 $y=y_1y_2\cdots y_{2n}$ 是否在 $\hat{L}$ 中的算法 B. 首先，确定 y 的形式是否是 $x_1x_2\cdots x_nx_1x_2\cdots x_n$. 这很容易检查. 出于讨论的缘故，假设回答此问题所花的时间为 0 秒. 如果 y 的形式不正确，则停止并回答 y 不在 $\hat{L}$ 中. 如果 y 的形式正确，则检查 y 的前 n 个数字是否构成 L 中的位串.
(a)计算算法 B 的最坏情况复杂度.
(b)计算算法 B 的平均情况复杂度.
(c)你的答案表明平均复杂度可能不是一个好的度量吗？为什么？

31

2.5 r 排列

给定一个 n 集合，假设我们想要挑选出 r 个元素，并依次排列它们. 这样的一个排列称为这个 n 集合的一个 r 排列(r-permutation of the n-set). $P(n, r)$将计数 n 集合的 r 排列的数量. 例如，计算没有重复字母的 3 字母单词的数量，可以从 26 个字母中选 3 个不同字母且依次排列它们完成；因此，我们有 $P(26, 3)$. 类似地，如果一个学生要完成 4 个实验且在 10 个学时中完成这些实验(每一个实验需要一个学时)，那么该学生为自己所能安排的不同的调度数量为 $P(10, 4)$. 注意，如果 $n<r$ 则 $P(n, r)=0$：在这种情况下不存在 n 集合的 r 排列. 在下文中，通常理解为 $n\geqslant r$.

为了弄清如何计算 $P(n, r)$，注意到在 3 字母单词的情况下，第一个字母存在26 种选择；对于其中的每一种选择，第二个字母存在 25 种选择；对其中的每一种选择，第三个字母存在 24 种选择. 因此，根据乘法规则有

$$P(26, 3)=26\times25\times24.$$

在实验调度的例子中，第一个实验有 10 种选择，第二个实验有 9 种选择，第三个实验有 8 种选择，第四个实验有 7 种选择，所以给出

$$P(10, 4)=10\times9\times8\times7.$$

根据同样的理由，如果 $n\geqslant r$[⊖]，则

$$P(n, r)=n\times(n-1)\times(n-2)\times\cdots\times(n-r+1).$$

如果 $n>r$，则上面的公式可以简化成如下形式：

$$P(n,r)=\frac{[n\times(n-1)\times\cdots\times(n-r+1)]\times[(n-r)\times(n-r-1)\times\cdots\times1]}{(n-r)\times(n-r-1)\times\cdots\times1}.$$

因此，我们可以得到如下结果：

$$P(n,r)=\frac{n!}{(n-r)!}. \tag{2.1}$$

我们在 $n>r$ 的假设下，得到(2.1)式. 显然对于 $n=r$ 也成立(为什么).

【例 2.19 CD 播放器】 我们新买一台有很多好功能的 CD 播放器. 特别地，这台播放器有标记为1～5 的 5 个 CD 槽，可按此顺序播放 CD. 如果我们的藏品中有 24 张 CD，那么为了视听娱乐，有多少种放入播放器槽的不同方法？对于第一个槽有 24 种选择，第二个槽有 23 种选择，第三个槽

32

⊖ 这一公式甚至对 $n<r$ 也成立. 为什么？

有22种选择，第四个槽有21种选择，第五个槽有20种选择，所以给出如下结果：

$$P(24, 5)=24\times 23\times 22\times 21\times 20.$$

另外，使用等式(2.1)，我们再一次看到

$$P(24,5)=\frac{24!}{(24-5)!}=\frac{24!}{19!}=24\times 23\times 22\times 21\times 20.$$ ■

本节练习

1. 计算：

   (a)$P(3, 2)$ (b)$P(5, 3)$ (c)$P(8, 5)$ (d)$P(1, 3)$

2. 设$A=\{1, 5, 9, 11, 15, 23\}$.

   (a)求由A的元素构成的长度为3的序列的个数.

   (b)假设A的每个元素都不使用两次，重复(a).

   (c)假设序列的第一个元素是5，重复(a).

   (d)假设序列的第一个元素是5且A中的元素都不使用两次，重复(a).

3. 设$A=\{a, b, c, d, e, f, g, h\}$.

   (a)求由A的元素构成的长度为4的序列的个数.

   (b)假设没有重复字母，重复(a).

   (c)假设序列的第一个字母是b，重复(a).

   (d)假设序列的第一个字母是b，最后一个字母是d，且没有重复字母，重复(a).

4. 如果我们需要安排20名工作申请人面试，那么我们有多少种不同的顺序安排前五个人的面试.

5. 如果一个校园的电话分机有4个数字，如果不重复使用数字，那么有多少部分机：

   (a)如果第一个数字不能是0?

   (b)如果第一个数字不能是0且第二个数字不能是1?

6. 一种常用的组合[⊖]锁(密码锁)或组合挂锁在刻度盘上有40个数字，范围从0到39. 通过顺时针、逆时针再顺时针转动刻度盘，并每次在特定数字停止以打开锁. 一个公司能够制造多少种不同的密码挂锁? [33]

2.6 子集

【例2.20 比萨问题】 一个比萨店做广告，说它可提供500多种比萨. 当地消费者保护署对此表示怀疑. 在这家比萨店，下面浇头(馅料)的任意组合都有可能成为一种比萨：

意大利辣香肠，蘑菇，胡椒粉，橄榄，香肠，凤尾鱼，意大利腊肠，洋葱，咸肉.

这家比萨店在它的广告中说实话了吗? 通过简单地运用乘法规则我们可以回答这一问题. ■

为了回答例2.20中的问题，考虑集合$\{a, b, c\}$. 我们问这个集合有多少个子集? 通过枚举法可以得到答案，我们发现存在8个这样的子集：

$$\varnothing, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}.$$

使用乘法规则也可以得到答案. 我们考虑按步构建子集. 首先，考虑包含或不包含元素a的情况. 这存在两种选择. 然后，考虑包含或不包含b的情况. 这也存在两种选择. 最后，是包含c或不包含c的情况. 仍有两种选择. 根据乘法规则，构建子集的总的方法数量为

$$2\times 2\times 2=2^3=8.$$

类似地，4集合的子集的数量是

$$2\times 2\times 2\times 2=2^4=16,$$

且n集合的子集数量是

⊖ 在2.7节，我们将看到词“组合”对挂锁并不合适：“r排列”是正确的.

$$\underbrace{2\times2\times\cdots\times2}_{n\text{次}}=2^n.$$

这些考虑对比萨问题有帮助吗？我们可以把一个特定比萨考虑成比萨浇头集合的一个子集. 另外，对于每一种浇头，我们可以考虑包含它或不包含它. 总之，我们看到存在 $2^9=512$ 种可能的比萨. 因此，这家比萨店没有做虚假广告.

本节练习

1. 枚举集合$\{a, b, c, d\}$的 16 个子集.
2. 一个杂志订阅服务经营 35 种杂志. 一位订户可以订阅其中任意多种杂志. 这一杂志订阅服务尝试用计算机处理它的付款过程，并希望给两名不同的人指定不同的计算机键(标识成员)，除非他们订阅的杂志完全相同. 这需要多大存储量，即需要多少种不同的代码数？

34

3. 如果例 2.20 的比萨店决定在它的比萨上总是放上洋葱和蘑菇，那么现在这家比萨店可以提供多少种不同的比萨？
4. 假设例 2.20 的比萨店增加一种新的浇头，沙丁鱼，但是强调说每一种比萨或者有沙丁鱼或者有凤尾鱼. 现在这家比萨店可以提供多少种可能的比萨？
5. 如果 A 是一个 10 元素集合，A 有多少非空子集？
6. 如果 A 是一个 8 元素集合，A 有多少元素个数大于 1 的子集？
7. 集合 A 上的*值函数*(value function)把 0 或者 1 指定给 A 的每一个子集.
   (a)当 A 有 3 个元素时，A 上有多少个不同的值函数？
   (b)当 A 有 n 个元素时情况如何？
8. 在一个简单游戏(参见 2.15 节)中，玩家的每一个子集被定义为赢或者输.
   (a)如果对这一定义没有限制，对于 3 位玩家有多少种不同的简单游戏？
   (b)对于 n 位玩家情况又如何？

2.7 r 组合

n 集合的一个 r 组合(r-combination of an n-set)是从这个集合中取出 r 个元素的一种选择，这意味着与顺序无关. 因此，一个 r 组合是一个 r 元素子集. $C(n, r)$将表示 n 集合的 r 组合的数量. 例如，从 4 个人中选出 3 个人组成一个委员会的方法数记为 $C(4, 3)$. 如果这 4 个人是 Dewey、Evans、Grange 和 Howe，那么可能的委员会是

{Dewey, Evans, Grange}, {Howe, Evans, Grange},
{Dewey, Howe, Grange}, {Dewey, Evans, Howe}.

因此，$C(4, 3)=4$. 我们将证明有关 $C(n, r)$的一些简单定理. 注意，如果 $n<r$ 则 $C(n, r)=0$. 在这种情况下，不存在 n 集合的 r 组合. 因此，通常认为 $n\geq r$. 本节中所有定理都假设 $n\geq r$.

定理 2.1

$$P(n,r) = C(n,r)\times P(r,r).$$

证明 n 个对象中的 r 个对象的有序排列可以通过下面的方式得到：首先选出 r 个对象(可以用 $C(n, r)$种方法完成)，然后再排列它们(这有 $P(r, r)=r!$ 种方法). 于是根据乘法规则上述定理成立. ■

推论 2.1.1

35

$$C(n,r) = \frac{n!}{r!(n-r)!}. \tag{2.2}$$

证明

$$C(n,r) = \frac{P(n,r)}{P(r,r)} = \frac{n!/(n-r)!}{r!/(r-r)!} = \frac{n!}{r!(n-r)!}.$$ ■

推论 2.1.2

$$C(n,r)=C(n,n-r).$$

证明

$$\frac{n!}{r!(n-r)!}=\frac{n!}{(n-r)!r!}=\frac{n!}{(n-r)![n-(n-r)]!}.$$ ■

对另一种“组合”证明，参见练习 20.

注意　数

$$\frac{n!}{r!(n-r)!}$$

通常记为

$$\binom{n}{r},$$

并称为二项式系数(binomial coefficient). 正如我们将看到的那样，这是因为这个数出现在二项式展开中(参见 2.14 节). 推论 2.1.2 说明如下结果：

$$\binom{n}{r}=\binom{n}{n-r}.$$

在下文中，我们将交替使用 $C(n, r)$和$\binom{n}{r}$.

定理 2.2

$$C(n,r)=C(n-1,r-1)+C(n-1,r).$$

证明　将 n 个对象中的一个用 * 标记. 选取的 r 个对象或者包含对象 * 或者不包含对象 *. 对于前者有 $C(n-1, r-1)$种方法，因为这相当于从 $n-1$ 个非 * 对象中选择 $r-1$ 个对象. 对于后者有 $C(n-1, r)$种方法，因为这相当于从 $n-1$ 个非 * 对象中取 r 个对象. 因此，根据加法规则得到这一定理. ■

注意　这一证明也可以陈述成“组合”证明，即依赖于计数论证. 这一定理也可以通过使用公式(2.2)进行代数处理而得到证明. 下面是一个“代数”证明.

36

定理 2.2 的第二种证明

$$\begin{aligned}C(n-1,r-1)+C(n-1,r)&=\frac{(n-1)!}{(r-1)![(n-1)-(r-1)]!}+\frac{(n-1)!}{r![(n-1)-r]!}\\&=\frac{(n-1)!}{(r-1)!(n-r)!}+\frac{(n-1)!}{r!(n-r-1)!}\\&=\frac{r(n-1)!}{r!(n-r)!}+\frac{(n-r)(n-1)!}{r!(n-r)!}\\&=\frac{r(n-1)!+(n-r)(n-1)!}{r!(n-r)!}\\&=\frac{(n-1)![r+n-r]}{r!(n-r)!}\\&=\frac{n!}{r!(n-r)!}\\&=C(n,r).\end{aligned}$$ ■

下面给出这些新公式及到目前为止的基本规则的简单应用.

1. 在比萨问题(例 2.20)中，正好有 3 种不同浇头的比萨的数量为

$$C(9,3)=\frac{9!}{3!6!}=84.$$

2. 根据加法规则，至多带有 3 种不同浇头的比萨数量为

$$C(9,0)+C(9,1)+C(9,2)+C(9,3)=130.$$

3. 在一个实验中，如果我们有 6 种药物需要测试，且我们想要从中选出 2 种给特定的对象，完成这一工作的方法数量是

$$C(6,2)=\frac{6!}{2!4!}=15.$$

4. 如果有 7 个可能的会议时间且一个委员会必须开 3 次会，我们可以安排会议时间的方法数量是

$$C(7,3)=\frac{7!}{3!4!}=35.$$

5. 从 9 个人中选出 5 人组成委员会的数量是

37

$$C(9,5)=126.$$

6. 从美国参议院中选出 7 个成员组成委员会的数量是

$$C(100,7).$$

7. 由 2 名参议员和 2 名众议员组成的总统代表团的数量是

$$C(100,2)\times C(435,2).$$

8. 带有 5 个 1 和 4 个 0 的 9 个数字位串的数量是

$$C(9,5)=C(9,4).$$

为了弄清为什么，考虑有 9 个未知数字，从中选择 5 个设为 1(或从中选择 4 个为 0).

计算数 $C(n, r)$的一个便利的方法是使用图 2.4 所示的数组. 数 $C(n, r)$出现在第 n 行，第 r 条对角线上. 给定位置上的每个元素可以通过把该位置的上一行中左边和右边的两个元素相加得到. 例如，$C(5, 2)$可以通过把图 2.4 中画圈的两个数 4 和 6 相加得到. 图 2.4 的数组被称为*帕斯卡三角形*(Pascal's triangle)，以法国著名的哲学家和数学家 Blaise Pascal 命名. 帕斯卡是概率论的创始人之一，他还发现了很多有趣的组合数学技术.

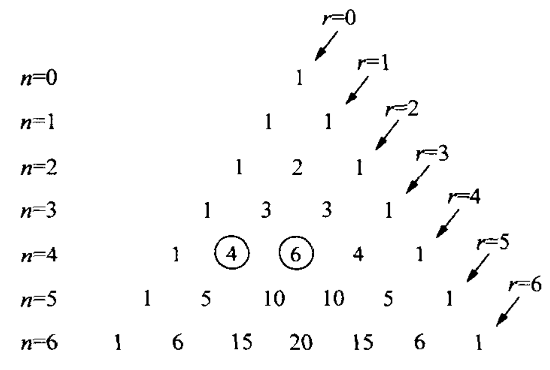


图 2.4　帕斯卡三角形. 圆圈中的数相加给出 C(5, 2)

为什么帕斯卡三角形正确？答案是它依赖于下面的关系

$$C(n,r)=C(n-1,r-1)+C(n-1,r). \tag{2.3}$$

这正好是我们在定理 2.2 中已证明的关系. 关系式(2.3)是*递推关系*(recurrence relation)的一个例子. 我们将在本书以后的章节中多次看到这样的关系式，特别在第 6 章. 第 6 章通篇讨论这一论题. 得到这样的关系式使我们可以把复杂的数值计算归约到早前的步骤，因此使得这些数值的计算得以分步进行.

本节练习

1. 从有 10 名队员的篮球队里选出 5 名首发队员(与位置无关)，有多少种选择方法?

2. 从有 50 名提名者中选出 7 名优胜者，有多少种选择方法?

3. 计算：

(a)$C(6, 3)$　(b)$C(7, 4)$　(c)$C(5, 1)$　(d)$C(2, 4)$

4. 计算 $C(n, 1)$.

5. 计算 $C(5,2)$ 并使用枚举法检查你的答案.

6. 计算 $C(6,2)$ 并使用枚举法检查你的答案.

38

7. 通过计算检查下式：

   (a) $C(7,2)=C(7,5)$　　(b) $C(6,4)=C(6,2)$

8. 通过加入新的一行来扩展图 2.4.

9. 计算 $C(5,3)$，$C(4,2)$ 和 $C(4,3)$ 并验证公式(2.3)成立.

10. 对 $C(7,5)$，$C(6,4)$ 和 $C(6,5)$ 重复练习 9.

11. (a)如果有 8 种血样，分成两组且每组有 4 种血样送往不同的实验室检测，那么有多少种分法？假设实验室是可区分的.

    (b)如果有 8 种血样，分成两组且每组有 4 种血样送往不同的实验室检测，那么有多少种分法？假设实验室是不可区分的.

    (c)如果有 8 种血样，分成两组且在每一组中至少有一种血样，那么有多少种分法？假设实验室是可区分的.

12. 一家公司正在考虑 6 种可能的新计算机系统，且它的系统管理员希望至多测试其中的 3 个系统. 系统管理员有多少种方法选择测试系统？

13. (a)把 10 种食物分成两组，且每组有 5 种食物送往不同的实验室检查其品质，有多少种分法？

    (b)把 10 种食物分成两组且每一组至少有一种食物，有多少种分法？

14. 使用字母表中的 26 个字母构成 8 字母单词，且没有重复字母，如果每个单词含有 3、4 或 5 个元音，那么能够构成多少个这样的单词？

15. 1000 到 9999 之间有多少个每位数字不同的奇数？

16. 一个车队是从 7 辆不同品牌的外国车和 4 辆不同品牌的国产车的集合中选出的. 对于下面的要求，计算车队的组合方法数量.

    (a)这个车队有 5 辆车，其中有 3 辆外国车和 2 辆国产车.

39

    (b)这个车队的大小是随意的(但不能没有车)，但是其中的外国车和国产车的数量必须相等.

    (c)这个车队有 4 辆车而且其中一辆必须是雪佛兰汽车.

    (d)这个车队有 4 辆车，外国车和国产车各有 2 辆车，而且雪佛兰和本田不能都在其中.

17. (a)一个计算机中心有 9 个不同的程序要运行，其中的 4 个程序使用 C++语言而 5 个程序使用 JAVA 语言. C++程序之间没有区别，JAVA 语言也没有区别. 对于下面的条件，确定程序运行的可能顺序数量.

    i. 没有限制.

    ii. C++程序必须相继运行.

    iii. C++程序必须相继运行且 JAVA 程序也必须相继运行.

    iv. 这些语言必须交替.

    (b)假设从 C++配置切换到 JAVA 配置的成本为 10 个单位，从 JAVA 配置切换到 C++配置的成本是 5 个单位，从 C++到 C++或从 JAVA 到 JAVA 的切换成本为 0. 运行这些程序的最高效(最小成本)顺序是什么？

    (c)如果 C++程序之间是可区分的，且 JAVA 的程序之间也是可区分的，重复(a).

18. 某个公司有 30 名女性雇员，其中有 3 名属于管理层，并有 150 名男雇员，其中有 12 名属于管理层. 选出一个由 3 名女性和 3 名男性组成的委员会. 对于下面的条件，有多少种组成这样的委员会的方法.

    (a)每种性别至少有 1 名管理层人员.

    (b)至少有 1 名管理层人员.

19. 考虑下面的恒等式

$$\binom{n}{m}\binom{m}{k}=\binom{n}{k}\binom{n-k}{m-k}.$$

    (a)使用“代数”证明形式证明这一恒等式.

(b)使用"组合"证明形式证明这一恒等式.

20. 通过使用 $C(n, r)$的定义给出推论 2.1.2 的另一种"组合"证明.

21. 如何从帕斯卡三角形求$\binom{n}{0}+\binom{n}{1}+\binom{n}{2}+\cdots+\binom{n}{n}$? 当 $n=2, 3, 4$ 时求这个和. 推测一般结果.

22. 证明

$$\binom{n}{0}+\binom{n+1}{1}+\cdots+\binom{n+r}{r}=\binom{n+r+1}{r}.$$

40

23. 证明下面恒等式. 该恒等式称为范德蒙德等式(Vandermonde's identity)(尽量使用"组合"证明方法).

$$\binom{n+m}{r}=\binom{n}{0}\binom{m}{r}+\binom{n}{1}\binom{m}{r-1}+\binom{n}{2}\binom{m}{r-2}+\cdots+\binom{n}{r}\binom{m}{0}.$$

24. 根据 Cohen[1978], 定义$\left\langle {n \atop r} \right\rangle$为$\binom{n+r-1}{r}$. 证明

$$\left\langle {n \atop r} \right\rangle=\left\langle {n \atop r-1} \right\rangle+\left\langle {n-1 \atop r} \right\rangle.$$

(a)使用代数证明方法　　(b)使用组合证明方法

25. 设$\left\langle {n \atop r} \right\rangle$的定义如练习 24 所示, 证明:

$$\left\langle {n \atop r} \right\rangle=\frac{n}{r}\left\langle {n+1 \atop r-1} \right\rangle=\frac{n+r-1}{r}\left\langle {n \atop r-1} \right\rangle.$$

26. 数的序列 $a_0, a_1, a_2, \cdots, a_n$ 称为**单峰的**(unimodal), 如果对于某个整数 t 有 $a_0\leqslant a_1\leqslant\cdots\leqslant a_t\leqslant$且$a_t\geqslant a_{t+1}\geqslant\cdots\geqslant a_n$. (注意, 帕斯卡三角形的任意行上的项先递增然后再递减, 所以形成一个单峰序列.)

(a)证明: 如果 $a_0, a_1, a_2, \cdots, a_n$ 为单峰的, 那么 t 不一定是唯一的.

(b)证明: 如果 $n>0$, 那么序列$\binom{n}{0}, \binom{n}{1}, \binom{n}{2}, \cdots, \binom{n}{n}$, 是单峰的.

(c)证明(b)中的序列的最大项是$\binom{n}{\lfloor n/2\rfloor}$, 其中$\lfloor x\rfloor$是小于或等于 x 的最大整数.

2.8 概率

组合数学的历史与概率论的历史几乎是缠绕在一起的. 不确定事件的研究发展了概率论, 即这些事件可能发生也可能不发生. 特别地, 这一理论是由帕斯卡、费马和拉普拉斯以及其他人通过对某些博弈结果的研究而发展起来的. 在拉普拉斯于 1812 年出版的《概率的理论分析》一书中, 他如下定义了概率: 一个事件的**概率**(probability)是表明该事件发生的可能结果的数量除以可能结果的总数. 例如, 假设我们考虑随机选出一个 2 数字位串. 这样的位串有 4 种: 00, 01, 10 和 11. 有 0 的位串的概率是多少呢? 答案是$\frac{3}{4}$. 因为表明问题中的事件发生的可能结果的数量是 3, 即有 0 的位串的数量, 而且总共有 4 种可能的结果. 正如我们很快就会看到的那样, 拉普拉斯的定义只适合于所有可能的结果均等时的情况.

41

我们来把问题阐述得更精确一些. 通过考虑产生多个可能结果中的一个结果的实验来尝试公式化概率的概念. 可能结果的集合称为**样本空间**(sample space). 一个**事件**(event)对应于结果集合的一个子集, 即样本空间的一个子集, 它对应于表明这一事件已经发生的那些结果. 一个事件的**补**(complement)对应于表明这一事件还没有发生的那些结果. 拉普拉斯的定义表明, 如果 E 是样本空间 S 的一个事件且 E^c 是 E 的补, 那么

$$E\text{ 的概率}=\frac{n(E)}{n(S)}\text{ 且 }E^c\text{ 的概率}=\frac{n(S)-n(E)}{n(S)}=1-\frac{n(E)}{n(S)},$$

其中 $n(E)$是 E 中的结果数量，$n(S)$是 S 中的结果数量．注意，由上式可知，E 的概率是 0 和 1 之间的一个数．

下面将这一定义应用于博弈．我们掷一个骰子，这是我们的实验．要计算结果是偶数的概率．样本空间是可能结果的集合{1，2，3，4，5，6}．问题中的事件是偶数所组成的结果集合，即集合{2，4，6}．那么我们有

$$\text{偶数的概率} = \frac{n(\{2,4,6\})}{n(\{1,2,3,4,5,6\})} = \frac{3}{6} = \frac{1}{2}.$$

注意，只有在样本空间中的所有结果都是均等时这一结果才成立．如果我们有一个经过加重、总出现 1 的骰子，那么得到偶数的概率不是 1/2 而是 0[⊖]．

考虑有两个孩子的家庭．这个家庭至少有一个男孩的概率是多少？组建这样的有两个孩子的家庭有 3 种可能：有两个男孩，两个女孩或者一个女孩和一个男孩．取这三种可能情况作为我们的样本空间．第一和第三个结果构成事件"至少有一个男孩"，因此，根据拉普拉斯的定义有

$$\text{至少有一个男孩的概率} = \frac{2}{3}.$$

这一结果真的是正确的吗？不．如果我们观察有两个孩子的家庭，至少有一个男孩的概率大于 $\frac{2}{3}$．这是因为有 4 种方法组建有两个孩子的家庭：我们可以先有一个男孩，然后有另外一个男孩、先有一个女孩，然后再有另外一个女孩、先有一个男孩然后再有另外一个女孩，或者先有一个女孩然后再有一个男孩．因此，有一个男孩和一个女孩的情况多于有两个男孩的情况，所以我们的样本空间中的结果不是均等的．然而，使用简洁的缩写形式表示，结果 BB，GG，BG 和 GB 是均等的[⊜]，所以我们取这四个结果作为我们的样本空间．现在"至少有一个男孩"的事件在我们的 4 个结果中有 3 个，所以我们有

42

$$\text{至少有一个男孩的概率} = \frac{3}{4}.$$

在本书，我们将局限于样本空间中结果均等情况的概率计算．注意，我们的概率定义只适用于样本空间有限的情况．对于样本空间无限的情况，很显然拉普拉斯的定义需要修改．对于不均等的情况和无限的情况，读者可以参见任何一本关于概率论的教科书，例如 Feller[1968]、Parzen[1992]和 Ross[1997]．

让我们继续给出几个概率定义的应用．假设已知一个家庭有 4 个孩子．其中一半是男孩的概率是多少？答案不是$\frac{1}{2}$．为了得到答案，我们观察到这一样本空间是长度为 4 的所有 B 和 G 的序列；一个典型的这样的序列是 BGGB．正好有两个 B 的序列有多少个呢？有 4 种位置，而且其中两个必须选择 B．因此，存在 $C(4，2)$这样的序列．这样的序列总共有多少个呢？根据乘法规则，这样的序列有 2^4 个．因此，

$$\text{一半是男孩的概率} = \frac{C(4,2)}{2^4} = \frac{6}{16} = \frac{3}{8}.$$

读者可以尝试写出所有 16 种可能的结果，且注意到表示正好有 2 个男孩的事件数为 6.

⊖ 可以证明我们给出的这一概率的定义是"循环定义"，因为它依赖于事件"均等"的概念，而这表明我们已经知道如何估计概率．这非常微妙．然而，我们可以在不能估计事物的情况下比较它们，例如，可以说这个人和那个人好像一样高．开始于这种比较的概率估计理论在 Fine[1973]和 Roberts[1976，1979]的著作中都有所描述．

⊜ 即使是这一陈述也并不完全精确，因为似乎生男孩的可能性略大于生女孩的可能性(参见 Cummings[1997])．因此，我们所选择的四个事件也不是完全均等的．特别地，BB 的可能性大于 GG 的可能性．然而，这一推断是一个良好的有效近似．

接下来，假设我们投掷5次硬币. 至少存在两次正面的概率是多少？样本空间是由长度为5的正面和反面的所有可能的序列组成的，用简洁的缩写形式表示即它是由HHHTH这样的序列组成的. 至少有两次是正面的这样的序列有多少？答案是存在$C(5,2)$个这样的序列正好有两个正面，$C(5,3)$个这样的序列正好有3个正面，以此类推. 因此，至少有两次正面的序列的数量由下式给出：

$$C(5,2)+C(5,3)+C(5,4)+C(5,5)=26.$$

可能的序列总数为$2^5=32$. 因此

$$\text{至少有两次正面的概率}=\frac{26}{32}=\frac{13}{16}.$$

【例2.21 系统的可靠性】 想象一个系统有n个组成部分，其中每一个组成部分或者运转或者失效. 如果第i个组成部分运转则令x_i为1，如果第i个组成部分失效则令x_i为0. 我们用位串$x_1x_2\cdots x_n$描述这一系统. 因此，位串0011描述一个有4个组成部分的系统，且前两个失效，而第三个和第四是运转的. 因为很多系统有固有的冗余，所以即使有一些组成部分失效，这一系统总体仍可以运转. 如果由$x_1x_2\cdots x_n$描述的这一系统运转，则令$F(x_1x_2\cdots x_n)$为1，否则令$F(x_1x_2\cdots x_n)$为0. 这时，F是一个从长度为n的位串到$\{0,1\}$的函数，即F是一个n变量开关函数(例2.4). 例如，假设我们有一个带有三个相同组成部分的高冗余系统，且这一系统运转，当且仅当至少有两个组成部分运转. 那么F就是由表2.4所给出的函数. 我们将在3.2.4节、练习22和13.3节中研究函数F的其他特殊例子. 假设一个系统中的组成部分运转和失效是均等的[⊖]. 那么任意两个位串同样都可能是描述这一系统的位串$x_1x_2\cdots x_n$. 现在我们可以问：这一系统运转的概率是多少，即$F(x_1x_2\cdots x_n)=1$的概率是多少？这就是这一系统的可靠性(reliability)的度量. 在我们的例子中，8个位串中的4个位串111，110，101和011是表明$F(x_1x_2x_3)=1$的事件. 因为所有位串都是均等的，所以系统工作的概率是$\frac{4}{8}=\frac{1}{2}$. 关于更多系统可靠性的计算法可以参见Karp和Luby[1983]、Barlow和Proschan[1975].

[43]

表2.4 开关函数F：当且仅当一个系统的两个或三个组成部分运转时其值为1

$x_1x_2x_3$	111	110	101	100	011	010	001	000
$F(x_1x_2x_3)$	1	1	1	0	1	0	0	0

系统可靠性理论因各种网络而得到广泛的研究：电网、计算机网络、通信网络和运输线路网络. 关于网络可靠性的更全面的参考可参见Hwang，Monma和Roberts[1991]、Ball，Colbourn和Provan[1995]. ■

在结束本节之前，对应于相关子集上的运算，我们注意到关于事件概率的一些通用陈述：

事件E不发生的概率是E^c的概率.

事件E或F发生的概率是$E\cup F$的概率.

事件E和F发生的概率是$E\cap F$的概率.

同样，从概率的定义容易看出

$$E^c\text{的概率}=1-E\text{的概率}. \tag{2.4}$$

如果E和F是不相交的，那么

$$E\cup F\text{的概率}=E\text{的概率}+F\text{的概率}. \tag{2.5}$$

[44]

更一般地，

⊖ 在更一般的分析中，我们将首先评估第i个组成部分运转的概率p_i.

$$E \cup F \text{的概率} = E \text{的概率} + F \text{的概率} - E \cap F \text{的概率}. \tag{2.6}$$

为了弄清为什么等式(2.6)为真，考虑图 2.5 中的维恩图. 注意，当我们把 E 的概率和 F 的概率相加时，我们把 E 和 F 的交的概率相加了两次. 通过把 E 和 F 的交的概率从它们的概率的和中减去，就可以得到等式(2.6).

为了说明这些观察结果，考虑投掷骰子的实验. 没有得到一个 3 的概率是 1 减去得到一个 3 的概率；即 $1-\frac{1}{6}=\frac{5}{6}$. 得到一个 3 或一个偶数的概率是多少呢？因为 $E=\{3\}$ 和 $F=\{2, 4, 6\}$ 不相交，(2.5)式表明它是 E 的概率加上 F 的概率，即 $\frac{1}{6}+\frac{3}{6}=\frac{2}{3}$. 最后，得到一个大于 4 的数或一个偶数的概率是多少？问题中的事件集合是 $\{2, 4, 5, 6\}$，其概率是 $\frac{4}{6}=\frac{2}{3}$. 注意这不等于大于 4 的数的概率与偶数的概率之和，即 $\frac{2}{6}+\frac{3}{6}=\frac{5}{6}$. 这是因为 $E=\{5, 6\}$ 和 $F=\{2, 4, 6\}$ 是相交的，$E\cap F=\{6\}$. 运用(2.6)式，我们有

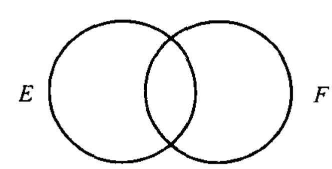


图 2.5　与等式(2.6)相关的维恩图

$E\cup F$ 的概率 $=\frac{2}{6}+\frac{3}{6}-\frac{1}{6}=\frac{2}{3}$，这与我们的第一个计算一致.

【例 2.22 食物过敏(再探例 2.5)】 在例 2.5 中，我们研究了与土豆、巧克力、玉米和花生这四种食物的某种组合引发食物过敏相关的开关函数. 我们已看到总共存在 16 种可能的食物组合. 我们考虑了在食物中有土豆或在食物中有玉米和花生时产生过敏反应的情况. 没有这样的过敏反应的概率是多少？

为了找到所求的概率，我们首先计算产生过敏反应的概率. 注意，如果食物被表示成位串 $(1, y, z, w)$ 或位串 $(x, y, 1, 1)$，则产生过敏反应，其中 x，y，z，w 是(二进制)0－1 值变量. 因为在第一种类型中存在三个二进制变量，而在第二种类型中存在两个二进制变量，所以第一种类型有 $2^3=8$ 种不同的位串，而第二种类型有 $2^2=4$ 种不同的位串. 如果两种类型之间没有重叠，那么(2.5)式允许我们把概率相加即可. 然而，当 x，z 和 w 都是 1 时，发生重叠. 在这种情况下，y 可以是 0 或 1. 因此，根据(2.6)式，产生过敏反应的概率是

$$\frac{8}{16}+\frac{4}{16}-\frac{2}{16}=\frac{10}{16}=\frac{5}{8}.$$

45

根据(2.4)式，不产生过敏反应的概率是 $1-\frac{5}{8}=\frac{3}{8}$. ■

在例 2.22 中，枚举出可能的食物组合是解决问题的一种有效技术. 然而，如果需要考虑的食物至少有 10 种，那么枚举出所有可能组合的方法将显得很笨拙. 因此，在本节所开发和使用的技术本质上要避免枚举法.

本节练习

1. 下列实验中的结果是均等的吗？

   (a)随机选出一位加利福尼亚市民并记录他的居住城镇.

   (b)把两个药丸和三个糖球放在一个容器内，随机取出一个并记录它的类型.

   (c)随机拾起一片雪花并记录它的形状.

   (d)掷两个骰子并记录出现的数字的和.

   (e)随机选取长度为 3 的位串，并观察它的数字和.

2. 掷一枚骰子，计算下面结果的概率：

(a)一个奇数　　(b)一个小于或等于 2 的数

(c)一个被 3 整除的数

3. 对于一个有 3 个孩子的家庭，计算下面各结果的概率：

(a)正好有 2 个男孩　　(b)至少有 2 个男孩

(c)至少有 1 个男孩且至少有 1 个女孩

4. 如果黑发、棕色发和金发是均等的(而且不会出现其他颜色的头发)，一个有 3 个孩子的家庭至少有两个孩子是金发的概率是多少？

5. 投掷一枚硬币四次，计算至多有三次正面的概率是多少？

6. 随机选取一个长度为 5 的 DNA 链，计算它至少有四个 A 的概率是多少？

7. 从一副 52 张的纸牌中随机抽取一张牌，这张牌是一张 K 或一张 Q 的概率是多少？

8. 从一副 52 张的纸牌中随机抽取一张牌，这张牌插回去后再随机抽取另外一张牌，得到两张 K 的概率是多少？

9. 随机选取一个长度为 4 的位串，至少有三个 1 的概率是多少？

46

10. 随机选取一个长度为 3 的位串，没有两个连续 0 的概率是多少？

11. 假设一个系统有 4 个独立的组成部分，每一个组成部分运转和失效的可能性都是均等的．假设当且仅当至少有 3 个组成部分运转时，这个系统才运转．这个系统运转的概率是多少？

12. 如果这个系统运转，当且仅当第四个组成部分运转且其他组成部分中至少有两个运转，重做练习 11.

13. 一个医学实验室能够运作的最低条件是有一名有执照的 x-射线技师和一名验血师．现有 3 名有执照的 x-射线技师和两名验血师，且每一名工作人员在指定日子是上班或呆在家里的可能性都是均等的．假设每一名工作人员都可独立决定是否来上班，这个实验室可以运作的概率是多少？

14. 假设我们有 10 双不同的手套．从 20 支手套中随机选出 4 支．至少得到一双手套的概率是多少？

15. 使用规则(2.4)～(2.6)计算将一枚硬币掷 6 次得到下面结果的概率：

(a)两次正面或三次正面

(b)两次正面或两次反面

(c)两次正面或在第一次投掷时是正面

(d)偶数个正面或至少 9 个正面

(e)偶数个正面或第一次投掷时是正面

16. 使用概率的定义证明规则：

(a)(2.4)　　(b)(2.5)　　(c)(2.6)

17. 假设只在下面的食谱中产生过敏反应，重复例 2.22 中的问题：

(a)或者包含土豆和玉米或者包含巧克力和花生

(b)或者包含土豆或者包含所有其他三种食物

2.9 放回取样

在美国国家曲棍球联合会(NHL)中，一个队在每次比赛中有赢、输和平局三种结果．在一个有 82 场比赛的日程表中，每个队可以有多少种不同的赛季战况[⊖]？根据乘法规则，答案是 3^{82}．对于这 82 场比赛的每一场，都存在 3 种可能性：赢、输或平局．我们说我们正在*放回取样*(sampling with replacement)．我们在选择一个 3 集合{赢，输，平局}的 82 排列，且当集合中的元素被抽出后再把这些元素放回集合．等价地，我们允许重复．设 $P^R(m, r)$ 是一个 m 集合的 r 排列的数量，允许放回或重复．那么由乘法规则得到如下的结果：

$$P^R(m,r) = m^r. \tag{2.7}$$

47

如果我们不能放回取样或重复取样，那么数 $P(m, r)$ 计数一个 m 集合的 r 排列的数量.

对于 r 组合的情况我们可以做类似的区分．设 $C^R(m, r)$ 是放回取样或重复取样时一个 m 集

⊖ 不要把“赛季战况”与“记录”混淆．记录指的是赢、输和平局的最终总和，而赛季是计数得到每一种记录的不同方法的数量.

合的 r 组合的数量．例如，在允许放回取样的情况下，2 集合 $\{a, b\}$ 的 4 组合可以得到下面的结果：

$$\{a, a, a, a\}, \{a, a, a, b\}, \{a, a, b, b\}, \{a, b, b, b\}, \{b, b, b, b\}.$$

因此，$C^R(2, 4)=5$．下面给出 $C^R(m, r)$ 的公式．

定理 2.3

$$C^R(m,r) = C(m+r-1,r).$$

我们将在本节末证明定理 2.3．现在用一些例子来说明它．

【例 2.23 巧克力店】 假设一家巧克力店有 3 种巧克力糖：樱桃(c)，桔子(o)和香草(v)．这家商店允许顾客选出一打巧克力装一个巧克力盒．有多少种不同的巧克力盒？我们可以认为有一个 3 集合 $\{c, o, v\}$，并从中选出一个 12 组合，可放回取样．因此，巧克力盒的数量是

$$C^R(3, 12)=C(3+12-1, 12)=C(14, 12)=91.$$ ■

【例 2.24 DNA 串：伽莫夫编码】 在 2.1 节中，我们研究了字母表 $\{A, C, G, T\}$ 上的 DNA 串，及为了编码一个氨基酸所需要的这样的串的最小长度．我们注意到有 20 种不同的氨基酸，如 2.1 节所示的那样，仅存在 16 种长度为 2 的 DNA 串，所以需要长度至少为 3 的串．但是存在 $4^3=64$ 种不同的长度为 3 的位串．伽莫夫[1954a，b]指出，氨基酸与 DNA 双链结构中由基形成的呈现菱形的“洞”之间存在联系．每一个菱形(参见图 2.6)由 4 个基组成，每一个基位于这个菱形的一角．我们将使用从这一菱形的顶点开始且按顺时针绕该菱形放置的 4 基序列 $xyzw$ 区分每一个菱形．例如，图 2.6 中的菱形可以写成 GTTA．由于在 DNA 中基是成对出现的，所以序列中的第四个基 w 总是由这个序列中的第二个基 y 来决定的．如果第二个基是 T，那么第四个基就是 A(反之亦然)，或者如果第二个基是 G，那么第四基是 C(反之亦然)．

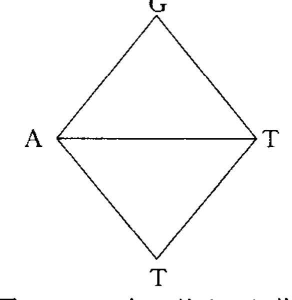


图 2.6 一个 4 基 DNA 菱形

伽莫夫指出，菱形 $xyzw$ 编码的氨基酸与(a) $xwzy$ 和(b) $zyxw$ 编码的氨基酸相同．因此，GTTA、GATT、TTGA 和 TAGT 都可编码相同的氨基酸．如果伽莫夫的意见是正确的，那么使用 4 基 DNA 菱形可以编码多少种氨基酸呢？在 4 基序列 $xyzw$ 中对于 y-w 对只存在 2 种选择：A-T(或同样的 T-A)和 G-C(或同样的 C-G)．挑选出其他两个基是定理 2.3 的应用．我们有 $m=4$ 个对象，希望可挑选出 $r=2$ 个对象，可放回取样，且与顺序无关．可按下式计算：

$$C(m+r-1, r)=C(5, 2)=10,$$

48

即有 10 种不同的组合．因此，使用伽莫夫 4 基 DNA 菱形正好可以编码 $2\times10=20$ 种不同的氨基酸．遗憾的是，后来发现这不是编码工作的方式．参见 Golomb[1962]的讨论．也可参见 Griffiths 等[1996]． ■

表 2.5 概括了我们讨论的放回取样和无放回取样．

表 2.5 从 m 元素集合中挑选 r 个元素的取样

是否与顺序有关	是否允许重复	取样名称	选择取样的方法数	参考
否	否	r 组合	$C(m, r)=\frac{m!}{r!\,(m-r)!}$	推论 2.1.1
是	否	r 排列	$P(m, r)=\frac{m!}{(m-r)!}$	式(2.1)
否	是	r 放回组合	$C^R(m, r)=C(m+r-1, r)$	定理 2.3
是	是	r 放回排列	$P^R(m, r)=m^r$	式(2.7)

【例 2.25 投票方法】 在美国的大多数选举中，若干候选人竞选一个公职，而且每一个已登记的投票人为他选择的候选人投票. 选举的胜利者是拥有最高投票总数的候选人(在选票相等的情况下，可能有多名胜利者，然后进行平局决赛). 这一投票方法称为*相对多数投票法*(plurality voting). 假设 3 名学生竞选一个有 400 名学生班级的学生班级主席. 如果每一个人都投票，有多少种不同的可能结果? 由于“不同的结果”，我们要提到这 3 名候选人得到的选票总数的不同“模式”的数量. 一个*模式*(pattern)是一个序列(a_1，a_2，a_3)，其中 a_i($i=1$，2，3)是候选者 i 所得到的票数. 因此，(6，55，339)与(55，339，6)是不同的结果，且(6，55，338)是不可能的. [我们不区分每一名投票人(例如，谁投了谁的票)，对于每一名候选人只以投票总数区分.]这又是定理 2.3 的一个应用. 我们有 $m=3$ 个对象，且我们希望选出 $r=400$ 个对象，(显然)可以放回取样. 通过下面的计算，我们可以得到完成这一任务的方法数量：

49

$$C(m+r-1,\ r)=C(402\ 400)=80\ 601.$$

这一答案假设每个投票人都投票. 练习 10 提出了当并非所有投票人都必须投票时的投票总数问题.

另一种投票方法称为*累积投票法*(cumulative voting)，它可以用于需要选出的候选人大于 1 的情况. 这就是许多市议会、董事会和教育委员会的选举情况(累积投票被用于从 1870 年到 1980 年的伊利诺伊州的立法委员会选举). 使用累积投票法，投票人的投票数与要填充的公开席位的数量一样多，而且不限制他们把所有选票投给一名候选人. 相反，他们可以对一名或多名候选人投复数票. 在有 p 名候选人、q 个公开席位和 r 名投票人这样的选举中，总共有 qr 可能投票总数. 类似于相对多数投票的情况，胜利者是 q 名有最大投票总数的候选人. 再一次考虑 3 名候选人和 400 名投票人的例子. 然而，现在假设学生不是在选举一名班级主席而是选出两名副主席. 根据累积投票方法，可能有多少种不同的投票总数? 正如上面的相对多数投票的例子一样，如果要求每一名投票人至少投票一名候选人，那么至少有 400 次投票且应该至多投出 $2\times400=800$ 张票. 考虑有 j 张投票的情况，其中 $400\leqslant j\leqslant800$. 根据定理 2.3，有

$$C^R(3,\ j)=\binom{3+j-1}{3-1}=\binom{2+j}{2}$$

种不同的投票总数. 因为 j 的范围是从 400 到 800，利用加法规则，有

$$\binom{2+400}{2}+\binom{2+401}{2}+\cdots+\binom{2+800}{2}=75\ 228\ 001$$

种不同的投票总数. 可不投票的累积投票法和其他的投票方法将在练习中给出. 对于投票方法以及相关数学的一般介绍可以参见 Aumann 和 Hart[1998]、Brams[1994]、Brams 和 Fishburn[1983]、Farquharson[1969]或 Kelly[1987]. ■

定理 2.3 的证明[㊀] 假设 m 集合有元素 a_1，a_2，…，a_m，那么这些对象的 r 个任意样本可以描述为其中 a_1 有多少个，a_2 有多少个等. 例如，如果 $r=7$，$m=5$，代表性的样本是 $a_1a_1a_2a_3a_4a_4a_5$ 和 $a_1a_1a_1a_2a_4a_5a_5$. 我们也可以通过在最后的 a_i，$i=1$，2，…，$m-1$ 之后画一条竖线表示这些样本. 这样，上面的两个样本可以写成 $a_1a_1\mid a_2\mid a_3\mid a_4a_4\mid a_5$ 和 $a_1a_1a_1\mid a_2\parallel a_4\mid a_5a_5$，其中，因为在第二种情况中没有 a_3，我们画了两条连续的竖线. 现在，如果我们使用这一记法描述 r 个对象的样本，那么我们可以省去下标. 例如，$aa\mid aa\mid\mid\mid aaa$ 则表示 $a_1a_1\mid a_2a_2\mid\mid\mid a_5a_5a_5$. 于是 r 个对象的样本数量恰好等于 r 个字母 a 和 $m-1$ 条竖线的不同排列数量. 这样的排列有 $m+r-1$ 个元素，而且我们通过为 a 选择 r 个位置来确定这个排列. 因此有 $C(m+r-1,\ r)$种这样的排列. ■

50

㊀ 此证明可以跳过.

本节练习

1. 如果允许放回取样，计算：

   (a)2 集合的 4 排列　　　　(b)3 集合的 2 排列

   (c)2 集合的 4 组合　　　　(d)3 集合的 2 组合

2. 通过使用等式(2.7)或定理 2.3 检查练习 1 的答案.

3. 如果允许放回取样，计算：

   (a)3 集合的 7 排列　　　　(b)4 集合的 7 排列

4. 如果有 4 场音乐会，有多少种方法可以选择 8 张门票?

5. 如果有 5 种不同的糕点，有多少种方法可以从中选出 12 块糕点?

6. 假设一个长度为 8 的代码字是由字母 A，B，C 或数字 0，1 组成的，且不以 1 开始. 有多少个这样的代码字?

7. 假设一个 DNA 链至少包含每一个基 T，C，A 和 G，长度为 6 的 DNA 链有多少? 根据下面的假设回答这一问题.

   (a)只考虑给定类型的基的数量.

   (b)考虑顺序.

8. 在一个有 82 场比赛的 NHL 的赛季中，有多少种可能的不同最终记录[⊖]?

   (a)如果每一场比赛一支球队或者赢或者输或者平局.

   (b)如果不能平局.

9. 美国的职业体协足球中，点球取胜和非点球取胜之间是有区别的. 所以有取胜、点球取胜、平局或输. 有多少种可能的 12 场比赛赛季?

10. 有 m 名候选人，n 名投票人，每一名投票人不必一定投票. 使用相对多数投票法(参见例 2.25)计算不同投票总数的数量.

11. 有 m 名候选人，n 名投票人，l 个公开席位，每一名投票人不必一定投票. 使用累积投票法(参见例 2.25)计算不同投票总数的数量.

2.10　分装问题[⊜]

2.10.1　分装问题的类型

在组合数学和概率论的发展历史中，把球放入盒子或小缸里的问题起着非常重要的作用. 这样的问题称为分装问题(occupancy problems). 分装问题有着广泛的应用. 根据偶然事件在一周内发生的日期分类这些偶然事件时，球是偶然事件的类型而盒子是这一周的各天. 在宇宙射线的实验中，球是到达盖格计数器的粒子，而盒子是计数器. 在编码理论中，可以通过对以代码字为盒子，以误差为球的研究，得到 k 个代码字上的传输误差的可能分布. 在图书出版中，k 页上误印的可能分布可以通过对以页码为盒子，以球作为误印的研究而得到. 在生物学的照射研究中，撞击视网膜的光粒子对应于小球，视网膜的细胞对应于盒子. 在礼券收集中，小球对应于特定的礼券，而盒子对应于礼券的类型. 我们将在本书的不同地方回答这些应用. 对于其他应用可以参见 Feller[1968, pp. 10-11]. 51

在分装问题中，是否认为两个小球是可区分的，以及是否认为两个盒子是可区分的将会产生很大的不同. 例如，假设我们有两个可区分的小球 a 和 b，以及三个可区分的盒子 1，2 和 3. 那么小球对盒子的可能分配如表 2.6 所示. 有 9 种不同的分配. 然而，假设有两个不可区分的球. 我们可以把它们都标为 a. 那么对 3 个可区分的盒子的可能分配如表 2.7 所示. 正好存在

⊖ 参见 32 页的脚注.

⊜ 为了快速阅读本节，阅读 2.10.1 节就足够了.

6 种分配．类似地，如果盒子是不可区分的，但是球是可区分的，表 2.6 的 1～3 的分配可以认为是相同的：两个球在一个盒子内，其他盒子里没有球．类似地，4～9 的分配可以认为是相同的：两个盒子各放一个球，另外一个盒子没有球．因此，只有两种不同的分配．最后，如果球和盒子都是不可区分的，那么表 2.7 中 1～3 的分配可以认为是相同的，而 4～6 的分配也是相同的，所以存在两种不同的分配．

根据是否允许盒子为空来区分分装问题也很常见．例如，如果我们有两个可区分的球和两个可区分的盒子，那么可能的分配如表 2.8 所示．存在 4 种不同的分配．然而，如果没有盒子是空的，那么只存在 2 种不同分配，即表 2.8 中的分配 3 和 4．

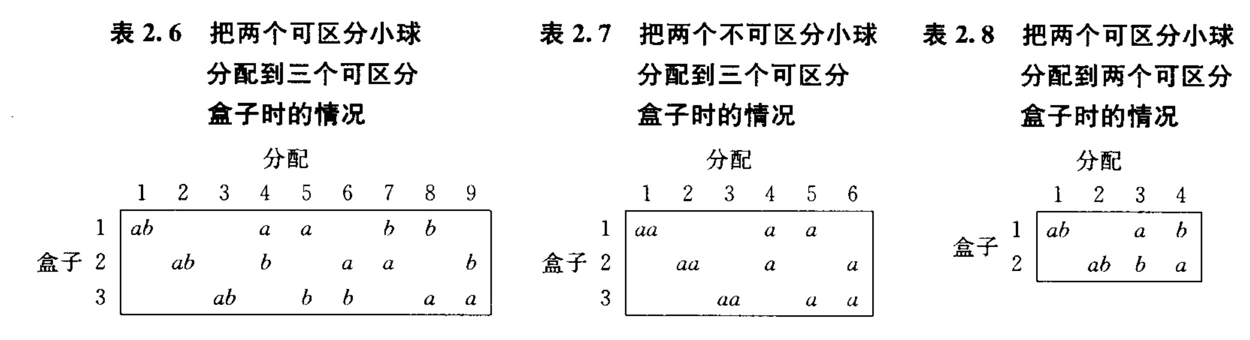


表 2.6　把两个可区分小球分配到三个可区分盒子时的情况

盒子 \ 分配	1	2	3	4	5	6	7	8	9
1	ab			a	a		b	b	
2		ab		b		a	a		b
3			ab		b	b		a	a

表 2.7　把两个不可区分小球分配到三个可区分盒子时的情况

盒子 \ 分配	1	2	3	4	5	6
1	aa			a	a	
2		aa		a		a
3			aa		a	a

表 2.8　把两个可区分小球分配到两个可区分盒子时的情况

盒子 \ 分配	1	2	3	4
1	ab		a	b
2		ab	b	a

表 2.9 概括了分装问题的可能情况．第四列中的某些记法和术语现在还没有定义，我们将在下面给出定义．下面将讨论各种情况．

52

表 2.9　分装问题分类

	是否可区分球	是否可区分盒子	盒子是否可空	把 n 个球放入 k 个盒子的方法数
情况 1				
1a	是	是	是	k^n
1b	是	是	否	$k!S(n, k)$
情况 2				
2a	否	是	是	$C(k+n-1, n)$
2b	否	是	否	$C(n-1, k-1)$
情况 3				
3a	是	否	是	$S(n, 1)+S(n, 2)+\cdots+S(n, k)$
3b	是	否	否	$S(n, k)$
情况 4				
4a	否	否	是	把 n 划分成 k 或更少个部分的划分数
4b	否	否	否	把 n 刚好划分成 k 个部分的划分数

2.10.2　情况 1：可区分球和可区分盒子

情况 1a 适合乘法规则：每个球可以选择 k 种盒子．如果 $k=3$ 且 $n=2$，我们得到 $k^n=9$，这就是表 2.6 所示的分配数量．我们在 2.10.4 节讨论情况 1b．

2.10.3　情况 2：不可区分球和可区分盒子[⊖]

由于下面的结果，情况 2a 遵循定理 2.3．

⊖　2.10 节其余部分可以省略．

定理 2.4 n 个不可区分的球放到 k 个可分区盒子的分配方法数量是 $C(k+n-1, n)$.

证明 假设盒子被标上标签 C_1, C_2, …, C_k. 可以通过列出每一个球到它所进入的盒子，来总结球放入到盒子的分配. 这时，一个分配对应于允许重复的 n 个盒子的集合(n 元多重集合). 例如，在表 2.7 中，分配 1 对应于集合 $\{C_1, C_1\}$，分配 5 对应于集合 $\{C_1, C_3\}$. 如果有 4 个球，集合 $\{C_1, C_2, C_3, C_3\}$ 对应于把一个球放入到 C_1，把一个球放入到 C_2，把两个球放入到 C_3 的分配. 因为一个分配对应于一个集合 $\{C_{i_1}, C_{i_2}, \cdots, C_{i_n}\}$，所以球放到盒子的分配方法与 k 集合 $\{C_1, C_2, \cdots, C_k\}$ 允许重复的 n 组合数量相同. 于是，由定理 2.3，其结果是 $C(k+n-1, n)$. ■ [53]

用表 2.7 解释定理 2.4. 我们有 $k=3$, $n=2$, 且 $C(k+n-1, n)=C(4, 2)=6$. 情况 2b 的结果是从情况 2a 的结果得到的. 给定 n 个不可区分球和 k 个可区分盒子，我们首先在每一个盒子里放一个球. 有一种方法完成这一工作. 还剩下 $n-k$ 个不可区分的球. 我们希望把这些球放入到 k 个可分区的盒子里，且没有盒子非空的限制. 根据定理 2.4，完成这一工作的方法数是

$$C(k+(n-k)-1, n-k) = C(n-1, k-1).$$

现在我们使用乘法规则得到表 2.9 中情况 2b 的结果. 注意，如果 $n<k$ 则 $C(n-1, k-1)$ 为 0. 不存在把 n 个球放入到 k 个盒子内且每一个盒子至少有一个球方法.

2.10.4 情况 3：可区分球和不可区分盒子

接下来考虑情况 3b. 设 $S(n, k)$ 被定义为把 n 个可区分球分配到 k 个不可区分盒子里，且没有空盒子的方法的数量. 数 $S(n, k)$ 被称为第二类斯特林数(Stirling number of the second kind)[⊖]. 在 5.5.3 节中我们将给出下面的公式：

$$S(n,k) = \frac{1}{k!}\sum_{i=0}^{k}(-1)^i\binom{k}{i}(k-i)^n. \tag{2.8}$$ [54]

为了说明这一结果，考虑情况 $n=2$, $k=2$. 那么有

$$S(n,k) = S(2,2) = \frac{1}{2}[2^2 - 2\cdot 1^2 + 0] = 1.$$

把两个可区分球 a 和 b 分配到两个不可区分盒子中，且每个盒子至少有一个球的方法只有一种：每个盒子中有一个球.

根据加法规则，现在从情况 3b 的结果可得到情况 3a 的结果. 为了把 n 个可区分球分配到 k 个不可区分盒子中且没有盒子是空的，或者一个盒子不是空的，或者两个盒子不是空的或者……现在，情况 1b 的结果也成立，因为把 n 个可区分球分配到 k 个可区分盒子中且没有空盒子可以这样实现：先把 n 个可区分的球放入 k 个不可区分盒子且没有空盒子[可以用 $S(n, k)$ 种方法实现]，然后再为这些盒子加标签(可以用 k 种方法实现). 例如，如果 $k=n=2$，那么根据我们前面的计算，$S(2, 2)=1$. 因此，把两个可区分球放入到两个可区分的盒子且没有空盒子的方法数是 $2!\,S(2, 2)=2$. 这就是早前我们从表 2.8 观察到的结果.

2.10.5 情况 4：不可区分球和不可区分盒子

为了处理情况 4a 和 4b，我们把一个正整数 n 的划分(partition)定义为和等于 n 的正整数的集合. 例如，整数 5 有如下划分：

$\{1, 1, 1, 1, 1\}$, $\{1, 1, 1, 2\}$, $\{1, 2, 2\}$, $\{1, 1, 3\}$, $\{2, 3\}$, $\{1, 4\}$, $\{5\}$.

注意，我们认为 $\{3, 2\}$ 与 $\{2, 3\}$ 相同. 我们所感兴趣的是在集合中有什么整数，而不关心它们在集合中的顺序. n 个不可区分球分配到 k 个不可区分盒子中的方法数显然与把整数 n 划分成至少

⊖ 第一类斯特林数存在并被用于其他研究中. 参见 3.4 节的练习 25.

k 个部分的方法数相同. 这就给出表 2.9 中情况 4a 的结果. 例如, 如果 $n=5$, $k=3$, 那么存在 5 种可能的划分, 即上面列出的后 5 种划分. 如果 $n=2$ 且 $k=3$, 那么存在两种可能的划分{1, 1}和{2}. 这一结果对应于表 2.7 中的两个不同分配: 两个盒子中各有一个球, 或一个盒子中有两个球. 表 2.9 中情况 4b 的结果也与此类似: 把 n 个不可区分球分配到 k 个不可区分盒子中且没有空盒子的方法数, 显然与把整数 n 正好划分成 k 个部分的方法数相同. 为了解释这一结果, 如果 $n=2$ 且 $k=3$,则没有方法划分 n.

我们将在练习中研究整数划分, 并在 5.3 节和 5.4 节的练习中重新讨论它们, 在那里我们将使用生成函数的方法处理它们. 对于划分的更详细讨论, 可以参阅大多数数论书籍, 例如, Niven[1991]或 Hardy 和 Wright[1980], 也可参阅 Berge[1971]或 Riordan[1980].

55

2.10.6 例子

现在, 我们给出几个运用表 2.9 的结果的例子. 读者应该注意到球或盒子是否可区分通常是判断的关键, 这依赖于我们对问题的解释和关注的内容.

【例 2.26 医院分娩】 假设一家医院 9 月份有 80 名婴儿出生, 并记录了每个婴儿的出生日期. 这一事件可能发生的方式有多少种? 婴儿视为球而日期视为盒子. 如果在两个婴儿之间不做区分但对日子要做区分, 那么处于情况 2, 其中 $n=80$, $k=30$, 这时, 上述问题的答案由 $C(109, 80)$给出. 如果只计数每天至少有一个婴儿出生这一事件发生的方法数, 其结果可以由 $C(79, 29)$给出. 如果只关心有两个婴儿出生、有三个婴儿出生的天数等, 而不关心具体的日子, 我们处于情况 4, 而且需要考虑把整数 80 划分成 30 或更少部分. ■

【例 2.27 编码理论】 在编码理论中, 首先把信息编码成已编码信息, 然后通过传输通道发送. 传输通道可能是电话线, 也可能是无线电波. 由于噪声或弱信号的缘故, 在接收的代码字中可能发生错误. 于是, 接收的代码字必须被解码成原来的信息. (第 10 章将介绍强调编码理论的密码技术.)

在监测传输通道的可靠性过程中, 假设我们保留发生错误的记录. 假设通过一条传输通道传输了 100 个已编码信息, 且发生了 30 处错误. 这一事件发生的方法有多少种? 错误是球而代码字是盒子. 似乎应该忽视错误之间的差异而集中精力关注在传输的某个时间段(由于外部因素或一个较高的负荷期)是否有更多的错误发生. 于是, 代码字被区分开来. 因此, 我们处于情况 2, 且答案由 $C(129, 30)$给出. ■

【例 2.28 性别分布】 假设我们记录得到计算机科学学位的前 1000 个人的性别. 人对应于球, 而两种性别对应于盒子. 我们一定要区分盒子. 如果我们区分人, 即如果区分诸如第 1 个人是男性、第 2 个人是男性等, 那么我们处于情况 1. 然而, 如果我们只对每一种性别的人的个数感兴趣, 那么我们处于情况 2. 对于前面的情况, 可能的分布数量是 2^{1000}. 对于后面的情况, 可能的分布数量由 $C(1001, 1000)=1001$ 给出. ■

【例 2.29 试演】 导演为一次个人百老汇演出的 8 家不同的旅游公司召集了 24 名演员(在需要替换演员的情况下要为旅游公司选出多个演员). 这些演员对应于球而旅游公司对应于盒子. 如果我们感兴趣的是在同一个旅游公司的演员, 那么我们可以考虑可区分球和不可区分盒子. 因为每一家旅游公司至少需要一名演员, 没有盒子是空的. 因此, 我们处于情况 3. 可能的分配数量由 $S(24, 8)$给出. ■

56

【例 2.30 统计力学】 在统计力学中，假设我们有一个 t 个粒子的系统. 假设有 p 个不同的状态或等级(例如，能量等级)，每一个粒子都在其中一个等级中. 我们可以通过给出粒子到等级的分布来描述这一系统的状态. 总之，如果粒子是可区分的，那么有 p^t 种可能的分布. 例如，如果我们有 4 个粒子和 3 个等级，那么有 $3^4=81$ 种不同的分布. 其中的一种分布是粒子 1 在等级 1 上，粒子 2 在等级 3 上，粒子 3 在等级 2 上，粒子 4 在等级 3 上. 另一种分布是粒子 1 在等级 2，粒子 2 在等级 1，粒子 3 和 4 在等级 3 上. 如果我们认为粒子到等级的任意分布的几率是均等的，那么任意给定排列的概率是 $1/p^t$. 在这样的情况下，我们说粒子遵循麦克斯韦-玻尔兹曼统计(Maxwell-Boltzmann statistics). 可是，似乎没有已知的物理粒子遵循麦克斯韦-玻尔兹曼统计；p^t 种不同的分布不是均等的. 事实上，对于很多不同粒子，在特殊的光子和核子中，改变一个相对简单的假设可以产生一个实验上的精确模型. 即假设我们认为这些粒子是不可区分的. 那么我们处于情况 2：粒子到各等级的两种分布被认为是相同的，如果相同的粒子数被分布到相同的等级上. 因此，上述的两种分布被认为是相同的，因为每一种分布都是把一个粒子分布到等级 1，一个粒子分布到等级 2，两个粒子分布到等级 3. 根据定理 2.4，分布 t 个粒子到 p 个等级的可区分方法的数量由 $C(p+t-1,t)$ 给出. 如果我们认为粒子到等级的任意分布是均等的，那么任意一个排列的概率是

$$\frac{1}{C(p+t-1,t)}.$$

在这一情况下，我们说这些粒子满足*博泽-爱因斯坦统计*(Bose-Einstein statistics). 如果我们认为粒子是不可区分的并假设在一个给定等级没有多于两个粒子存在，那么就得到统计力学中的第三个模型. 于是，我们得到*费米-狄拉克统计*(Fermi-Dirac statistics)(参见练习 21). 对于我们所描述的所有情况的更详细讨论，可以参阅 Feller[1968]或 Parzen[1992]. ■

本节练习

提醒读者注意：当不清楚球或盒子是否可区分时，你应该陈述你的解释，给出其理由，然后继续你的工作.

1. 写出下面各种情况的所有分配：
   (a)把 3 个可区分球 a，b，c 分配到 2 个可区分盒子 1，2.
   (b)把 4 个可区分球 a，b，c，d 分配到 2 个可区分盒子 1，2.
   (c)把 2 个可区分球 a，b 分配到 4 个可区分盒子 1，2，3，4.
   (d)把 3 个不可区分球 a，a，a 分配到 2 个可区分盒子 1，2.
   (e)把 4 个不可区分球 a，a，a，a 分配到 2 个可区分盒子 1，2.
   (f)把 2 个不可区分球 a，a 分配到 4 个可区分盒子 1，2，3，4.

57

2. 在练习 1 中，如果盒子是不可区分的，那么哪些分配是不同的?
3. 使用表 2.9 的结果，计算练习 1 中每一种情况的分配数量，并通过比较得出的结果，检查你的答案.
4. 如果盒子是不可区分的，重复练习 3.
5. 使用表 2.9 的结果，计算练习 1 的每一种情况中没有空盒子的分配数量. 并通过比较得出的结果，检查你的答案.
6. 如果盒子是不可区分的，重复练习 5.
7. 求下面整数的所有划分：
   (a)4　　(b)5　　(c)8
8. 求下面各种情况的划分：
   (a)把 9 分成 4 个或更少部分　　(b)把 11 分成 3 个或更少部分

9. 计算：

(a) $S(n, 0)$　(b) $S(n, 1)$　(c) $S(n, 2)$　(d) $S(n, n-1)$　(e) $S(n, n)$

10. 在检查校对者的工作时，我们要查看一篇文稿中的 5 种误印，有多少种方法可以找到 12 个误印？
11. 在练习 10 中，假设我们不区分误印的类型，但是要记录误印出现的页码．我们可以用多少种不同的方法找到出现于 75 页文稿中的 25 个误印？
12. 在例 2.27 中，假设我们查明 30 种错误而且要查明这些错误是否倾向于出现在一起，但不关心它们一起出现在哪个代码字中．如果已知每一种错误只在某一代码字中出现一次，有多少种方法找到 100 个代码字中的 30 种错误？
13. 有 9 名乘客的电梯在 5 个不同层上停留．如果我们只对一起离开的乘客感兴趣，有多少种可能的分布方法？
14. 让激光瞄准 5 个肿瘤，用 10 束激光去撞击，有多少种方法(不必假设每束激光撞击一个肿瘤)？
15. 盖格计数器记录一个时间周期内 6 种不同放射性粒子的碰撞．要得到 30 个计数有多少种不同的方法？
16. 把 10 名顾客分配给 7 名推销员，使得每一名推销员至少得到一名顾客的分配方法有多少种？
17. 把 10 名学生成对分在实验室工作，有多少种不同的分配方法？
18. 把 6 项工作指定给 4 名工人，使得每一项工作都有一个工人，且每一个工人至少得到一项工作的分配方法有多少种？

[58]

19. 把一个有 20 个元素的集合恰好分成 4 个子集，有多少种划分方法？
20. 在例 2.30 中，假设有 8 个光子和 4 个能量等级，且在每一个能量等级上有 2 个光子．在粒子不可区分的假设下(博泽-爱因斯坦统计)，这一情况出现的概率是多少？
21. 证明在例 2.30 中，如果粒子是不可区分的，但没有两个粒子能在同一等级上，那么存在 $C(p, t)$ 种把 t 个粒子安排到 p 个等级上的可能排列(假设 $t\leqslant p$)．
22. (a)根据组合的论点证明：

$$S(n,k)=kS(n-1,k)+S(n-1,k-1).$$

(b)使用(a)的结果描述如何通过类似于帕斯卡三角形的方法，计算第二类斯特林数．

(c)运用(b)的结果计算 $S(6, 3)$．

23. 根据组合的论点证明：

$$S(n+1,k)=C(n,0)S(0,k-1)+C(n,1)S(1,k-1)+\cdots+C(n,n)S(n,k-1).$$

24. (a)如果在划分中考虑顺序，那么{3，2}不同于{2，3}．在考虑顺序的情况下，求 5 的划分数量．

(b)在考虑顺序的情况下，求把 5 正好划分成 2 部分的划分数量．

(c)在考虑顺序的情况下，证明把 n 正好划分成 k 部分的划分数量由 $C(n-1, k-1)$ 给出．

25. 贝尔数(Bell number) B_n 是把 n 元素集合划分成非空、不可区分盒子的划分数量．注意：

$$B_n=S(n,0)+S(n,1)+\cdots+S(n,n).$$

证明

$$B_n=\binom{n-1}{0}B_0+\binom{n-1}{1}B_1+\cdots+\binom{n-1}{n-1}B_{n-1}.$$

2.11　多项式系数

2.11.1　带有特殊分配的分装问题

本节考虑把 n 个可区分球分配到 k 个可区分盒子的分装问题．特别地，我们考虑把 n_1 个球分配到第一个盒子，把 n_2 个球分配到第二个盒子，…，把 n_k 个球分配到第 k 个盒子．设

$$C(n;n_1n_2,\cdots,n_k)$$

表示完成这一工作的方法数量．本节集中研究数 $C(n; n_1, n_2, \cdots, n_k)$，这个数有时写作

$$\binom{n}{n_1, n_2, \cdots, n_k},$$

并称为多项式系数(multinomial coefficient). 59

【例 2.31 学校注册】 大学教务处有这样一个问题. 它要把 11 名新生都安排到一个入门课程的 4 个组里：3 名学生安排到第一组，第二组和第三组要分别有 4 名学生，0 名学生在第四组(这一组已经满员). 有多少方法可能实现这一安排? 答案是 $C(11; 3, 4, 4, 0)$. 现在，第一组有 $C(11, 3)$种选择；对于其中的每一种选择，第二组有 $C(8, 4)$种选择；对于其中每一种选择，第三组有 $C(4, 4)$种选择；对于其中的每一种选择，第四组有 $C(0, 0)$种选择. 因此，根据乘法规则，对应各组的人数分配的方法数量是

$$\begin{aligned} C(11;3,4,4,0) &= C(11,3)\times C(8,4)\times C(4,4)\times C(0,0) \\ &= \frac{11!}{3!8!}\times\frac{8!}{4!4!}\times\frac{4!}{4!0!}\times\frac{0!}{0!0!} = \frac{11!}{3!4!4!}, \end{aligned}$$

因为 $0! = 1$. 当然 $C(0, 0)$总是等于 1，所以这一答案等于 $C(11; 3, 4, 4)$. 另外，$C(4, 4)=1$,所以答案也等于 $C(11, 3)\times C(8, 4)$. 因为一旦有 3 个学生选择了第一组且有 4 个学生已经选择了第二组，那么对第三组选择剩余的 4 个学生是唯一的方法.

注意如果 11 名学生的分组是随机的，那么存在 4^{11}种可能的安排：对于每一名学生，有 4 个组选择. 因此，3 名学生在第一组、4 名学生在第二组和第三组、0 名学生在第四组的概率由下式给出：

$$\frac{C(11;3,4,4,0)}{4^{11}}.$$

一般地，假设 $\Pr(n; n_1, n_2, \cdots, n_k)$表示 n 个球被随机分配到 k 个盒子里的概率，那么在盒子 i 中将存在 n_i 个球，$i=1, 2, \cdots, k$. 于是

$$\Pr(n;n_1,n_2,\cdots,n_k) = \frac{C(n;n_1,n_2,\cdots,n_k)}{k^n}.$$

(为什么?)注意当计算多项式系数时，空盒子的存在并不影响计算. 这是因为

$$C(n;n_1,n_2,\cdots,n_j,0,0,\cdots,0) = C(n;n_1,n_2,\cdots,n_j).$$

然而，多项式分配的概率要受到空盒子的影响，因为分母基于盒子的总数量，它是空盒子和非空盒子的总和.

继续我们的例子，假设突然第四组有了空额. 现在教务处希望在第一组、第二组、第三组都安排 3 个人，而第四组安排 2 个人. 有多少种方法可以实现? 11 名学生中，3 名学生必须选择第一组；剩余的 8 名学生中，3 名学生必须选择第二组；剩余的 5 名学生中，3 名学生必须选择第三组；最后，剩余的 2 名学生必须被安置到第四组. 这样的安排方法的总数是

$$\begin{aligned} C(11;3,3,3,2) &= C(11,3)\times C(8,3)\times C(5,3)\times C(2,2) \\ &= \frac{11!}{3!8!}\times\frac{8!}{3!5!}\times\frac{5!}{3!2!}\times\frac{2!}{2!0!} = \frac{11!}{3!3!3!2!}. \end{aligned}$$

■ 60

下面推导 $C(n; n_1, n_2, \cdots, n_k)$的公式. 根据例 2.31 所使用的类似推理，得到

$$\begin{aligned} C(n;n_1,n_2,\cdots,n_k) &= C(n,n_1)\times C(n-n_1,n_2)\times C(n-n_1-n_2,n_3)\times\cdots \\ &\quad\times C(n-n_1-n_2-\cdots-n_{k-1},n_k) \\ &= \frac{n!}{n_1!(n-n_1)!}\times\frac{(n-n_1)!}{n_2!(n-n_1-n_2)!}\times\frac{(n-n_1-n_2)!}{n_3!(n-n_1-n_2-n_3)!} \\ &\quad\times\cdots\times\frac{(n-n_1-n_2-\cdots-n_{k-1})!}{n_k!(n-n_1-n_2-\cdots-n_k)!} \end{aligned}$$

$$= \frac{n!}{n_1!n_2!\cdots n_k!(n-n_1-n_2-\cdots-n_k)!}.$$

因为 $n_1+n_2+\cdots+n_k=n$，且因为 $0!=1$，我们有下面的结果.

定理 2.5

$$C(n;n_1,n_2,\cdots,n_k)=\frac{n!}{n_1!n_2!\cdots n_k!}.$$

现在我们给出这一定理的几点说明和几个应用.

1. 由三个 1 和一个 0 组成的 4 数字位串的数量是 $C(4;3,1)$：从四个位置中，我们选三个位置放置数字 1，一个位置放置数字 0. 因此，这样的位串的数量由下式给出：

$$C(4;3,1)=\frac{4!}{3!1!}=4.$$

四个这样的位串是 1110，1101，1011，0111.

2. 由两个 2，两个 3 和一个 1 组成的 5 数字数的数量是

$$C(5;2,2,1)=\frac{5!}{2!2!1!}=30.$$

3. 注意 $C(n;n_1,n_2)=C(n,n_1)$，为什么这是正确的?

4. 一个 NHL 赛季由 82 场曲棍球比赛组成. 这一赛季以胜 41 场，输 27 场和平局 14 场结束的方式数量是

$$C(82;41,27,14)=\frac{82!}{41!27!14!}.$$

5. RNA 是由 DNA 定义其链接的信使分子. RNA 链在每一个链接处有四个基之一. 除基尿嘧啶(U)取代了基胸腺嘧啶(T)之外，可能的基与 DNA 中的可能基相同(例 2.2). 由 3 个胞核嘧啶(C)和 3 个腺嘌呤(A)组成的长度为 6 的可能 RNA 链有多少种? 为了回答这一问题，我们考虑链中的 6 个位置，把这些位置分成四个集合，3 个位置进入 C 集合，3 个位置进入 A 集合，0 个位置进入 T 和 G 集合. 实现这一分配的方法数量由下式给出：

61

$$C(6;3,3,0,0)=C(6;3,3)=20.$$

6. 长度为 6 的可能 RNA 链有 4^6 种. 如果 RNA 链是随机产生的(例如，所有可能性都是均等的)，那么得到一个有 3 个 C 和 3 个 A 的概率是

$$\Pr(6;3,3,0,0)=\frac{C(6;3,3,0,0)}{4^6}=\frac{C(6;3,3)}{4^6}=\frac{20}{4096}=\frac{5}{1024}\approx 0.005.$$

注意结果不是 $\Pr(6;3,3)$.

7. 由 3 个 A，2 个 C，2 个 U 和 3 个 G 组成的 10 链接 RNA 链的数量是

$$C(10;3,2,2,3)=25\,200.$$

8. 在上面例子中所描述的 RNA 链以 AAG 结束的数量是

$$C(7;1,2,2,2)=630,$$

因为现在只有前 7 个位置需要填充，2 个 A 和 1 个 G 已经用尽. 注意，知道一个链的结尾情况可以大幅度减少可能链的数量. 在下一节，我们将看到通过明智地使用分解 RNA 链的各种酶，如何进一步限制可能链的数量，再通过一定的探测性工作，我们就可以揭示原来的 RNA 链且不必实际对它进行观察.

2.11.2 带有不可区分对象类的排列

应用 1、2 和 5～8 给出下面的一般概念：假设有 n 个对象，类型 1 有 n_1 个，类型 2 有 n_2 个，…，

类型 k 有 n_k 个，且 $n_1+n_2+\cdots+n_k=n$. 假设相同类型的对象是不可区分的. 这些对象的可区分排列的数量记为 $P(n;\ n_1,\ n_2,\ \cdots,\ n_k)$. 我们在这里使用“可区分”一词，是因为我们假设相同类型的对象是不可区分的. 例如，假设 $n=3$，且有两个类型 1 的对象 a 和 a，一个类型 2 的对象 b. 那么存在三个对象的 $3!=6$ 种排列，其中有几个是不可区分的. 例如，两个 a 中的第一个 a 在第二个 a 的位置上的 baa 与两个 a 中的第二个 a 在第二个 a 的位置上的 baa 是不可区分的. 只有三种可区分的排列 baa，aba 和 aab. 62

定理 2.6

$$P(n;n_1,n_2,\cdots,n_k)=C(n;n_1,n_2,\cdots,n_k).$$

证明 在这一排列中我们有 n 个位置或地方需要填充，我们把其中的 n_1 个位置指定给类型 1 的对象，把其中的 n_2 个位置指定给类型 2 的对象，以此类推. ■

在 2. 13 节我们讨论带有不可区分对象类的排列.

本节练习

1. 计算：

   (a)$C(7;\ 2,\ 2,\ 2,\ 1)$　(b)$C(9;\ 3,\ 3,\ 3)$　(c)$C(8;\ 1,\ 2,\ 2,\ 2,\ 1)$

   (d)$\Pr(6;\ 2,\ 2,\ 2)$　(e)$\Pr(10;\ 2,\ 1,\ 1,\ 2,\ 4)$　(f)$\Pr(8;\ 4,\ 2,\ 2,\ 0,\ 0)$

   (g)$P(9;\ 6,\ 1,\ 2)$　(h)$P(7;\ 3,\ 1,\ 3)$　(i)$P(3;\ 1,\ 1,\ 1)$

2. 计算 $C(n;\ 1,\ 1,\ 1,\ \cdots,\ 1)$.

3. 计算：

   (a)$P(n;\ 1,\ n-1)$　(b)$\Pr(n;\ 1,\ n-1)$　(c)$\Pr(n;\ 1,\ n-1,\ 0)$

4. 一个代码是使用 5 种符号 $+$，$\#$，$\bowtie$，∇ 和 $\otimes$ 书写的.

   (a)每一种符号正好使用 2 次，有多少个 10 位代码字？

   (b)如果随机选取一个 10 位代码字，每个符号正好使用两次的概率是多少？

5. 在一个空间陕小的狗屋，12 条狗必须放入 3 个笼子，4 条狗放入笼子 1，5 条狗放入笼子 2，3 条狗放入笼子 3. 实现这一安排有多少种方法？

6. 使用三个符号 a，b，c 书写一个代码.

   (a)恰好使用 4 个 a，1 个 b 和 2 个 c 书写的 7 位代码字有多少？

   (b)如果随机选取一个 7 位代码字，正好使用 4 个 a，1 个 b 和 2 个 c 的概率是多少？

7. 使用五个数字 1，2，3，4，5 书写一个代码.

   (a)每一个数字正好使用 3 次的 15 位代码字有多少？

   (b)如果随机选取一个 15 位代码字，每一个数字正好使用 3 次的概率是多少？

8. 与如下链有相同基组成的 RNA 链有多少？

   UGCCAUCCGAC?

9. (a)使用单词 excellent 中的所有字母构成的不同“字”有多少种？

   (b)如果随机选取一个 9 字母的“字”，那么它使用单词 excellent 的所有字母的概率是多少？ 63

10. 从 4 个 a，4 个 b，4 个 c 和 4 个 d 中选择字母形成一个 10 字母序列，如果每一个字母至少出现两次，有多少种方法形成这样的序列？

11. 如果没有两个 a 是相邻的，那么字母 a，a，a，a，a，b，c，d，e 的可区分排列有多少个？

12. 在粒子是可区分的假设下(麦克斯韦-玻尔兹曼情况)重复 2. 10 节的练习 20.

13. (a)假设我们可以区分撞击到视网膜上的 5 个不同的光粒子. 把这些光粒子分配给三个细胞，且有三个粒子撞击到第一个细胞，有一个粒子撞击到其余的每一个细胞，这样的分配方法有多少种？

    (b)如果我们已知分配给三个细胞的 5 个不同的光粒子，它们以(a)情况分配的概率是多少？

14. 假设有 35 个放射性粒子撞击一个有 50 个计数器的盖格计数器. 除第 35 个粒子之外的全部粒子以多少

种不同方式撞击第一个计数器?

15. 假设有 6 个人接受工作面试.

(a)有多少种不同的方法使其中有 2 个人在星期一、2 个人在星期三、2 个人在星期六面试?

(b)给定 6 次面试,如果 6 个人的面试日期是随机指定的,那么(a)中面试的安排的概率是多少?

(c)如果面试被安排在 3 天进行,每天面试 2 个人,实现这一安排有多少种方法?

16. 假设我们有 4 个元素. 有多少种方法把这些元素分配到 4 个可区分集合,且每个集合分配一个元素? 假设这些元素分别是

(a)a, b, c, d　(b)a, b, b, b　(c)a, a, b, b　(d)a, b, b, c

2.12 酶的完全分解[⊖]

考虑不用实际观察给定 RNA 链本身,就可以发现其样子的问题(2.11 节介绍了 RNA 链). 有一些酶在每一个 G 链接后面把 RNA 链分解成片段. 而另一些酶则在每一个 C 或 U 链接后面把这个链分解成片段. 例如,假设我们有链

CCGGUCCGAAAG.

使用 G 酶把这个链分解成如下的片段:

64

G 片段:CCG,G,UCCG,AAAG.

于是我们知道有这些片段,但是不知道它们出现的顺序. 这四个片段组成多少可能的链呢? 答案是 4! =24 条链:对应于这些片段的每一个不同排列都有一条链. 一条这样的链(不同于原来的链)是

UCCGGCCGAAAG.

假设我们接下来使用 U、C 酶,它在每个 C 链接或 U 链接后面把一条链分解成片段. 我们得到下面的片段:

U、C 片段:C,C,GGU,C,C,GAAAG.

同样,我们知道有这些片段,但是我们不知道它们出现的顺序. 这些片段可组成多少条链? 有人想说有 6! 条这样的链,但是这不正确. 例如,如果片段是

C,C,C,C,C,C.

使用这些片段就不存在 6! 条链,而是仅存在一条链,这条链是

CCCCCC.

这是因为有些片段是不可区分的. 为了计数由给定的片段所形成的可区分链的数量,我们注意到存在六个片段. 其中有四个是 C 片段,一个是 GGU,一个是 GAAAG. 因此,根据定理 2.6,由这些片段组成的可能链的数量是

$$P(6;4,1,1)=C(6;4,1,1)=\frac{6!}{4!1!1!}=30.$$

实际上,这一计算还没有结束. 注意在 U、C 片段中,片段 GAAAG 不可能出现,除非做为终止片段,因为 GAAAG 不结束于 U 或 C. 因此,我们知道这条链结束于

GAAAG.

还有五个 U、C 片段:C,C,C,C 和 GGU. 由这些片段组成的链的数量是

$$C(5;\ 4,\ 1)=5.$$

把 GAAAG 加到以这 5 条链之一作为开始的链上得到可能的链. 可能的链是

CCCCGGUGAAAG,

⊖ 本节可以跳过去而不失连续性. 这一部分的内容直到 11.4 节才会再遇到. 然而,本节包含一个实用话题的详细讨论,且我们总是把这一话题置于我们的课程中.

CCCGGUCGAAAG，

CCGGUCCGAAAG，

CGGUCCCGAAAG，

GGUCCCCGAAAG.

65

我们还是没有把关于 G 片段和 U、C 片段的知识都结合起来. 使用我们的知识能够知道原来链的所有信息吗？我们所列出的 5 个链中哪一个有正确的 G 片段？第一条链不是正确的，因为它有一个 G 片段 CCCCG，这不可能出现在以 G 作酶的片段中. 类似的分析表明只有第三个链

CCGGUCCGAAAG

有正确的 G 片段集合. 因此，我们从那些给定的 U、C 片段中恢复了初始链.

这就是从给定的完全酶分解(complete enzyme digest)恢复 RNA 链的例子. 即：在每个 G 链接后有一个分解，在每一个 U 或 C 链接后有另一个分解. 值得注意的是，通过考虑片段的两个集合，我们已经能够把片段的任意集合的巨大数量的可能链限制到仅一个可能链. 如果我们考虑尝试推测仅知道链的基，但不知道基的顺序的链，那么结果更令人注目. 这时，我们有

$C(12;4,4,3,1)=138\,600$ 条可能的链！

考虑另一个例子. 假设我们知道一个 RNA 链在完全酶分解后生成下面的 G 和 U、C 片段：

G 片段：UG，ACG，AC，

U，C 片段：U，GAC，GAC.

我们能够发现原来的 RNA 链吗？首先，我们再一次问 U、C 片段能否告诉我们这个链的哪部分应当出现在最后？在这种情况下，答案是不能. 然而，G 片段能：作为 G 片段，AC 只能出现在最后. 因此，剩余的两个 G 片段可以按任意顺序排列，且具有给定 G 片段的可能链是

UGACGAC 和 ACGUGAC.

现在，上面的两个链的后者生成的 AC 为 U、C 片段之一. 因此，前者一定是正确的链.

知道 G 片段和 U、C 片段也不总能完全恢复原来的 RNA 链. 有时候，这两个酶的完全分解在下面的意义下是不明确的：存在带有相同的 G 片段集合和相同的 U、C 片段集合的两个 RNA 链. 我们要求读者作为练习证明这一点(练习 8).

本节所描述的“分段策略”是 R. W. Holley 和他在 Cornell 大学的工作伙伴用来(Holley 等[1965])确定第一个核酸序列时所使用的. 这一方法已不再使用，事实上，在其他更有效的方法引入之前它只使用了不长的一段时间. 然而，它却有着重大的历史意义，展示了数学方法在生物学中的重要作用. 今天，通过使用放射性标记和高速计算机分析，更快地排列长 RNA 链已成为可能.

66

关于完全酶分解的更详细内容，有兴趣的读者请参见 Hutchinson[1969]、Mosimann[1968]、Mosimann 等[1966]. 我们将在 11.4.4 节再次讨论这一问题.

本节练习

1. 经过完全分解后，一条 RNA 链有下面的 G 片段和 U、C 片段：

   G 片段：CUG，CAAG，G，UC，

   U，C 片段：C，C，U，AAGC，GGU.

   (a)有多少条 RNA 链有这些 G 片段？

   (b)有多少条 RNA 链有这些 U、C 片段？

   (c)确定所有具有这些 G 片段和 U、C 片段的 RNA 链.

2. 对于练习 1，确定与给定的 G 片段所组成的链有相同基的 RNA 链的数量.

3. 对于下面的 G 片段和 U、C 片段，重复练习 1：

G 片段:G,UCG,G,G,UU

U、C 片段:GGGU,U,GU,C

4. 在练习 3 中，确定与给定的 G 片段所组成的链有相同基的 RNA 链的数量.
5. 对于下面的 G 片段和 U、C 片段重复练习 1：

G 片段:G,G,CC,CUG,G,

U、C 片段:GGGC,U,C,GC.

6. 在练习 5 中，确定与给定的 G 片段所组成的链有相同基的 RNA 链的数量.
7. 一个位串在每一个 1 和每一个 0 之后被断开. 结果片段如下所示(不一定按正确的顺序排列)：

在 1 之后断开:0,001,01,01,

在 0 之后断开:0,10,0,10,10.

(a)在 1 之后断开的具有这些片段的位串有多少个?

(b)在 0 之后断开的具有这些片段的位串有多少个?

(c)确定有上述两种片段集合的所有位串.

8. 寻找一条在下面意义下不明确的 RNA 链：存在另一条带有相同 G 片段和 U、C 片段的链. (你能找到带有六个或较少链接的链吗?)
9. 在练习 8 的意义下不明确的最短的可能 RNA 链是什么?

67

10. 如果一个位串如练习 7 那样被断开，那么它是不明确的吗? 为什么?

2.13 再论带有不可区分对象类的排列

在 2.11 节和 2.12 节中，我们遇到计数一个对象集的排列数量的问题，且其中一些对象是不可区分的. 本节我们要开发计数这一状况的另一种方法.

【例 2.32 “热手效应”】 一名篮球队员发现在一次早前的比赛中，他的 10 次投篮有 4 次命中，6 次失球. 然而，所有的 4 次命中都在前面. 这名篮球队员的观察可以缩写成如下形式：

XXXXOOOOOO,

其中 X 表示命中，而 O 表示失球. 这一观察是一次偶然事件吗? 或表明这名队员有一个“热手效应(hot hand)”呢? 热手效应假设一旦一名队员命中一球，那么他下一球命中的几率就会高于平均几率(对于热手效应现象的详细分析参见 Tversky 和 Gilovich[1989]). 为了回答这一问题，假设没有诸如热手这样的情况，即在一个球命中(或失球)之后，下一个球并不会有更多(或更少)的可能命中. 因此，假设命中的发生是随机的，而且每一次投球命中的概率相同，与其他投球的发生状况无关. 即 4 个进球和 6 个失球的所有可能的顺序是均等的.㊀这样的顺序有多少种呢? 使用 2.11 节的记法，这一答案是

$$P(10;6,4)=C(10;6,4)=\frac{10!}{4!6!}=210.$$

为了直接得到这一结果，注意这里有 10 个位置，而且我们希望把其中的 4 个指定给 X，其中的 6 个指定给 O. 因此，这样的顺序数量是 $C(10;6,4)=210$. 如果所有这样的顺序是均等的，那么看到特殊排列

XXXXOOOOOO

的概率是 1/210. 当然，这是看到任意一个给定排列的概率. 更有趣的是计算 10 个投球中正好有 4 个命中，且它们是连在一起的概率. 在 10 个投球中有 4 个进球、6 个失球，4 个进球连在一

㊀ 这并不预示 10 个进球和失球的所有排列是均等的. 例如，即使投进一个球是随机发生的，如果投进一个球几乎不可能，那么序列 OOOOOOOOOO 比序列 XXXXXXXXXX 更有可能.

起的排列有多少种？为了回答这一问题，考虑把 4 个进球作为一个单位 X*．于是我们希望只考虑计算 1 个 X* 和 6 个 O 的顺序的数量．于是

$$C(7;1,6)=\frac{7!}{1!6!}=7$$

68

种这样的顺序．这些对应于下面的顺序：

XXXXOOOOOO （即 X* OOOOOO），
OXXXXOOOOO （即 OX* OOOOO），
OOXXXXOOOO （即 OOX* OOOO），
OOOXXXXOOO （即 OOOX* OOO），
OOOOXXXXOO （即 OOOOX* OO），
OOOOOXXXXO （即 OOOOOX* O），
OOOOOOXXXX （即 OOOOOOX*）．

给定 4 个进球和 6 个失球，看到 4 个进球和 6 个失球且所有进球连在一起的概率是 7/210，即$\frac{1}{30}$．这一概率相当小．因此，看到所有进球都连在一起很不寻常，我们期望这不是随机出现的，因此有了热手效应这样的事情(我们在练习 6.8 讨论热手效应的问题)．

在结束这一例子之前，我们重复一下 4 个进球和 6 个失球的排列方式的数量计算．假设我们为这些投球加标签使得它们是可区分的：

$$X_a,X_b,X_c,X_d,O_a,O_b,O_c,O_d,O_e,O_f.$$

这 10 个标签的排列数是 10!．对于每一个这样的排列，我们可以随意重排列 4 个 X；这样就有 4! 个重排列．就我们的讨论而言，每个重排列产生一个排列，可以认为是相同的．类似地，我们也可以用6! 种方法重排列 6 个 O．因此，有 4! ×6! 个排列是相同的，而且每一个排列都对应于 4! ×6! 个类似排列．不可区分排列数是

$$\frac{10!}{4!6!}.$$ ■

扩展例 2.32 的推理给出了 $P(n;n_1,n_2,\cdots,n_k)=C(n;n_1,n_2,\cdots,n_k)$(定理 2.6)的另一个证明．

本节练习

1. 假设一个研究人员发现 12 个陪替氏(petri)培养皿中，有 4 个已生长而有 8 个没有生长．已生长的这 4 个培养皿彼此相邻．假设 12 个培养皿中有 4 个已生长，且这些培养皿的所有排序都是均等的，4 个已生长的培养皿彼此相邻的概率是多少？
2. 一位市场调查员发现一个拍卖台上的 11 辆汽车中，4 辆是进口车，7 辆是国产车．4 辆进口车彼此相邻．假设 11 辆汽车中 4 辆是进口车，且这 11 辆车的所有排序都是均等的，那么这 4 辆进口车彼此相邻的概率是多少？
3. 假设一名林务官发现有一些树生病了(S)而有一些树很好(W)，他发现一行 13 棵树中，前 5 棵是 S，后 8 棵是 W.

69

   (a)假设疾病的发生是随机的，而且所有树生病的概率都相同，且与其他树的状况无关，观察到给定顺序的概率是多少？
   (b)13 棵树中有 5 棵生病，8 棵是好的，且 5 棵生病的树连在一起的概率是多少？
4. 假设一名林务官发现有一些树生病(S)，有一树很好(W)，且有一些树有疑问(Q)．假设有 30 个棵树，10 棵有病或有疑问，这 10 棵树连续出现的概率是多少？假设有病的、好的及 10 棵怀疑有病的树的所有序列都是均等的．

5. 假设在一个街区有 11 座房子排列在一起，有 6 座受到白蚁的侵扰.
   (a)这 6 座房子相邻的方式有多少种?
   (b)这 6 座房子都互不相邻的方式有多少种?
   (c)有多少种方法设计一个访问顺序，使得走访其中的两座房子且这两座房子没有白蚁?
   (d)有多少种方法设计一个访问顺序，使得走访其中的两座房子且至少有一座房子没有白蚁?
6. 一对夫妇生 9 个孩子且有 5 个男孩，有多少种不同的顺序使得首先生一个男孩且所有 4 个女孩是连续出生的?
7. 在一个圆桌边围座坐 8 个人，有多少种不同的就座方法?（理解“不同”在这里的意义）
8. 随机选取一个长度为 4 的 RNA 链，这一 RNA 链有下列情况的概率是多少?
   (a)至少有三个连续的 C.　　(b)至少有两个连续的 C.
   (c)有一个连续的 AG.　　(d)一个连续 AUC.
9. 长度为 21 的位串，每一个 1 后面跟着 0 且有十七个 0 和四个 1，这样的位串有多少种?
10. 长度为 20 的 RNA 链，有 5 个 A，4 个 U，5 个 C 和 6 个 G，且每个 C 后面跟着 G，这样的 RNA 链有多少条?
11. 有 20 个人，他们的记录被有序地存储在一个文件里. 我们想要随机地选出其中的 4 个人进行调查，并保证不选取连续的两个人. 有多少种方法实现? [提示：或选取最后一位，或者不.]

2.14 二项式展开

假设 6 位牌照是这样做成的：这六位中的每一位或者是一个数或者是一个字母. 回想一下加法和乘法规则，有

70

$$(10+26)^6 = 2\,176\,782\,336$$

种可能的牌照. 扩展这一情况，假设牌照需要 n 位且每一位或者是 a 个事物中一个，或者是 b 个事物中的一个. 那么有

$$(a+b)^n$$

种牌照. 作为本章所考虑的这一想法的一个运用，我们开发一个关于 $(a+b)^n$ 的有用公式.

定理 2.7(二项式展开) 对于 $n \geqslant 0$,

$$(a+b)^n = \sum_{k=0}^{n} C(n,k) a^k b^{n-k} = \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k}.$$

证明[⊖] 注意:

$$(a+b)^n = \underbrace{(a+b)(a+b)\cdots(a+b)}_{n\text{次}}.$$

在各项相乘过程中，我们从每一个因子 $(a+b)$ 中挑出一项. 因此，我们仅得到形如 $a^k b^{n-k}$ 的项. 为了求出 $a^k b^{n-k}$ 项的系数，注意到，为了得到 $a^k b^{n-k}$ 我们需要从 n 个 $(a+b)$ 中选出 k 个 a. 这一选取方法有 $\binom{n}{k}$ 种. ■

特别地，我们有

$$(a+b)^2 = \binom{2}{0}a^2 + \binom{2}{1}ab + \binom{2}{2}b^2 = a^2 + 2ab + b^2$$

$$(a+b)^3 = \binom{3}{0}a^3 + \binom{3}{1}a^2b + \binom{3}{2}ab^2 + \binom{3}{3}b^3 = a^3 + 3a^2b + 3ab^2 + b^3$$

⊖ 对于另一种证明，参见练习 12.

$$(a+b)^4 = \binom{4}{0}a^4 + \binom{4}{1}a^3b + \binom{4}{2}a^2b^2 + \binom{4}{3}ab^3 + \binom{4}{4}b^4$$
$$= a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4.$$

读者可以把这里的二项式系数与帕斯卡三角形(图 2.4)中出现的数相比较. 你注意到相似之处了吗?

不难把定理 2.7 的二项式展开扩展到$(a+b+c)^n$ 的展开，而且更一般地有

$$(a_1 + a_2 + \cdots + a_k)^n$$

的展开. 我们把这一扩展留给读者(练习 5 和 8).

下面给出若干二项式展开的应用. 通过在定理 2.7 中取 $a=1$, $b=x$, 可求出$(1+x)^{30}$在展开中 x^{20}的系数. 我们求的是 $1^{10}x^{20}$的系数，即为 $C(30, 10)$.

71

定理 2.8 对于 $n \geqslant 0$，有

$$\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n} = 2^n.$$

证明 注意：

$$2^n = (1+1)^n.$$

因此，根据 $a=b=1$ 时的二项式展开，有

$$2^n = \sum_{k=0}^{n} \binom{n}{k} 1^k 1^{n-k}. \qquad \blacksquare$$

还可以按如下观点考虑定理 2.8. 数 2^n 计数 n 集合的子集的数目. 定理 2.8 的左侧计数 n 集合的 0 元素子集数目，加上 1 元素子集数目，加上 2 元素子集数目，一直加上 n 元素子集的数目. 用这种方法每一个子集刚好计数一次.

定理 2.9 对于 $n>0$，有

$$\binom{n}{0} - \binom{n}{1} + \binom{n}{2} - \cdots + (-1)^k\binom{n}{k} + \cdots + (-1)^n\binom{n}{n} = 0.$$

证明

$$0 = (1-1)^n = (-1+1)^n = \sum_{k=0}^{n} \binom{n}{k} (-1)^k (1)^{n-k}. \qquad \blacksquare$$

推论 2.9.1 对于 $n>0$，有

$$\binom{n}{0} + \binom{n}{2} + \cdots = \binom{n}{1} + \binom{n}{3} + \cdots.$$

这一推论可以解释为：从 n 个对象中选取偶数个对象的方法数等于选取奇数个对象的方法数.

本节练习

1. 写出：

   (a)$(x+y)^5$　　(b)$(a+2b)^3$　　(c)$(2u+3v)^4$

2. 确定下列式子的展开式中 x^{11} 的系数：

   (a)$(1+x)^{16}$　　(b)$(2+x)^{14}$　　(c)$(2x+4y)^{11}$

3. $(1+x)^{12}(1+x)^4$的展开式中 x^9 的系数是什么?
4. $(1+x)^{10}(1+x)^6$ 的展开式中 x^8 的系数是什么?

72

5. 确定$(a+b+c)^n$ 的展开公式.
6. 确定$(a+b+c)^6$ 的展开式中 $a^2b^2c^2$ 的系数.
7. 确定$(x+y+z)^5$ 的展开式中 xyz^3 的系数.
8. 确定$(a_1+a_2+\cdots+a_k)^n$ 的展开公式.
9. $(a+b+c)^6$ 的展开式中 a^3bc^2 的系数是什么?

10. $(x+y+z+2w)^6$ 的展开式中 xy^2z^2w 的系数是什么？
11. $(a+5b+2c+2d)^{12}$ 的展开式中 $a^3b^2cd^6$ 的系数是什么？
12. 通过对 n 做归纳证明定理 2.7.
13. 求 $\binom{12}{0}+\binom{12}{2}+\binom{12}{4}+\binom{12}{6}+\binom{12}{8}+\binom{12}{10}+\binom{12}{12}$ 的值.
14. 证明一个 n 集合的偶数子集的个数等于 2^{n-1}.
15. 一个位串具有偶数奇偶性(even parity)，如果它有偶数个 1. 长度为 n 的位串中有多少个位串有偶数奇偶性？
16. 求：

(a) $\sum_{k=0}^{n} 2^k\binom{n}{k}$ (b) $\sum_{k=0}^{n} 4^k\binom{n}{k}$

(c) $\sum_{k=0}^{n} x^k\binom{n}{k}$ (d) $\sum_{k=2}^{n} k(k-1)\binom{n}{k}$

(e) $\sum_{k=1}^{n} k\binom{n}{k}$ [提示：对 $(x+1)^n$ 的展开式求微分，并设 $x=1$.]

17. 证明：

(a) $\sum_{k=1}^{n} k\binom{n}{k}2^{k-1}2^{n-k}=n(4^{n-1})$ (b) $\sum_{k=1}^{n} k\binom{n}{k}2^{n-k}=n(3^{n-1})$

2.15 简单游戏中的势力[⊖]

2.15.1 简单游戏的例子

本节把前面所描述的某些计数规则应用于多人游戏的分析中. 在现代应用数学中，游戏不仅仅意味着垄断游戏、国际象棋或纸牌游戏. 它可以是一组玩家为各种奖励或报酬而竞争的任意情况. 在这种意义下，政治也是一种游戏，经济市场也是一种游戏，国际贸易也是一种游戏等. 在这里我们将采用这种更广泛的游戏意义.

把一个游戏考虑成有 n 个玩家的集合 I. 我们感兴趣的是各玩家间的可能合作，并相应地研究玩家的联合(coalition)，它对应于集合 I 的子集. 我们集中讨论简单游戏(simple game)，其中的每一联合或者是赢或者是输. 我们可以通过给定一个值函数 v 来定义一个简单游戏，该值函数把数字 0 或 1 指定给每个联合 $S\subseteq I$，且当 S 是一个输联合时 $v(S)$ 等于 0，而当 S 是一个赢联合时 $v(S)$ 等于 1. 在游戏理论中，通常假设输联合的子集不能是赢联合，我们将做这一假设. 通常还假设对于所有的 S，或者 S 或者 $I-S$ 是输联合. 我们将做这种假设.

73

简单游戏的非常重要的例子是加权多数游戏. 在一局*加权多数游戏*(weighted majority game，又译为加权强对策)中，有 n 个玩家，玩家 i 有 v_i 张投票，一个联合是赢的，当且仅当它至少有 q 张投票. 我们把这一游戏记为

$$[q; v_1, v_2, \cdots, v_n].$$

或者 S 或者 $I-S$ 是输联合的假设对允许的 q，v_1，v_2，…，v_n 给出了限制. 例如，[3; 4, 4]不满足这一要求. 加权多数游戏发生在公司，在那里玩家是股东，而且一名股东对拥有的每一股都有一张投票. 大多数立法机构都是形如[q; 1, 1, …, 1]的加权多数游戏，其中每一个玩家有一张投票. 然而，有一些立法机构给立法者的投票数量对应于他的管区的人口. 例如，1964 年的纽约纳苏郡的郡议会就是加权多数游戏[59; 31, 31, 21, 28, 2, 2](Banzhaf[1965]). 另一个例子是欧洲联盟部长理事会. 这一组织由 15 个成员国组成，为联盟立法. 在大多数情况下，这个理事会通过由如下所示加权成员国进行“资格多数”投票来决策：

⊖ 这一节可以跳过而不失连续性. 阅读本节要求 2.1 节～2.8 节的知识.

德国，法国，意大利，英国	各 10 票
西班牙	8 票
比利时，希腊，荷兰，葡萄牙	各 5 票
奥地利，瑞典	各 4 票
爱尔兰，丹麦，芬兰	各 3 票
卢森堡	2 票
总计	87 票

资格多数至少要求有 62 票. 因此，这一加权多数游戏是

$$[62;10,10,10,10,8,5,5,5,5,4,4,3,3,3,2].$$

(关于这一标准加权多数游戏的一个有趣的修改是：在某些情况下，资格多数投票还是 62 票，但是这些票必须由至少 10 个成员国投票. 我们将不考虑这种情况.)

也许大多数初等加权多数游戏是[2; 1, 1, 1]. 在这一游戏中有三个玩家，每一个玩家都有一张投票，且玩家的简单多数形成一个赢联合. 因此，这个赢联合是集合

$$\{1,2\},\{1,3\},\{2,3\},\{1,2,3\}.$$

74

简单游戏的另一个例子是联合国安理会. 这里有 15 个玩家：5 个常任理事和 10 个非常任理事. 对于有关实质事项的决定，一个联合是赢联合，当且仅当它含有所有 5 个常任理事成员，因为他们有一票否决权[⊖]，而且至少有 10 个非常任理事成员中的 4 个成员(对于程序性事项，决策的通过至少需要 15 个成员中的 9 个成员的赞成票). 有趣的是，对于实质决策，可以把安理会看成一个加权多数游戏. 因为可以把这一游戏考虑成

$$[39;\ 7,\ 7,\ 7,\ 7,\ 7,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1],$$

其中前五名玩家对应于常任理事. 易见，这一游戏的赢联合恰好与安理会的常任理事相同. 因此，即使在安理会中非显式地指定加权投票，这个游戏还是可以认为是加权多数游戏. (请读者考虑如何得到这些数，如 39，7 和 1，从而把一个简单游戏变成加权多数游戏.)

澳大利亚政府也出现类似的情况. 在做国家决策时，6 个州和联邦政府都起作用. 事实上，一项措施通过，当且仅当它至少有 5 个州或至少 2 个州和联邦政府的支持. 易见，这一简单游戏(在有相同赢联合的意义下)对应于游戏[5; 1, 1, 1, 1, 1, 1, 3]，其中第七个玩家是联邦政府.

并非每一个简单游戏都是加权多数游戏. 两院立法委员会就是一个例子(参见练习 5).

2.15.2 Shapley-Shubik 势力指数

我们将讨论简单游戏中玩家的势力(power)的度量问题：他操纵赢联合的能力. 首先注意势力不一定与玩家拥有的投票数成比例. 例如，比较两个游戏[2; 1, 1, 1]和[51, 49, 48, 3]. 两个游戏都有 3 个玩家，而且 2 个或更多玩家的任意联合都是赢联合. 因此，在这两个游戏中，玩家 3 是在同样的赢联合中，因此实质上拥有相同的势力. 这两个游戏也许可以解释成有 3 个政党的立法委员会. 在第一个立法委员会中，有 3 个同等的政党，且 3 个中必须有 2 个赞同才能使一项措施通过. 在第二个立法委员会中，有 2 个大党和一个小党. 然而，假设政党成员作为一个团体进行投票，那么一项措施仍需要得到 3 个政党中 2 个政党的赞同才能通过，所以，从效果上第三个小政党与其在第一个立法委员会中的势力一样.

有许多度量简单游戏势力的可供选择的方法. 其中包括由 Banzhaf[1965]、Coleman[1971]、Deegan 和 Packel[1978]、Johnston[1978]以及 Shapley 和 Shubik[1954]提出的势力指数. 我们将

75

⊖ 这就是"强权一致"规则.

在 5.7 节中讨论 Banzhaf 和 Coleman 的势力指数. 前者已应用于法院的"一个人"、"一张投票"的情况. 我们集中研究 Shapley-Shubik 势力指数，该指数由 Shapley[1953]以原始形式提出，并由 Shapley-Shubik[1954]以我们提到的形式推广.（在更一般的原始形式中，它被称为 Shapley 值.）对于 Shapley-Shubik 指数的文献介绍参见 Shapley[1981]，对于关于势力度量的文献概览参见 Lucas[1983].

考虑通过一次增加一个玩家，直到达到一个赢联合来构建一个联合. 当一个玩家的加入导致输联合转变成赢联合时，这个玩家被称为**中轴**(pivotal). 更形式化地，考虑玩家的任意排列，如果 i 前面的玩家集合是输联合且 i 前面的玩家和 i 组成的集合是赢联合时，称玩家 i 为这一排列的中轴. 例如，在游戏[2; 1, 1, 1]中，玩家 2 在 1, 2, 3 的排列和排列 3, 2, 1 中是中轴. 在简单游戏中，玩家 i 的 Shapley-Shubik 势力指数的定义如下所示：

$$p_i=\frac{\text{玩家 } i \text{ 为中轴的玩家排列数}}{\text{玩家的排列数}}.$$

如果我们认为玩家的一个排列是随机选取的，那么玩家 i 的 Shapley-Shubik 势力指数是玩家 i 为中轴的概率. 例如，在游戏[2; 1, 1, 1]中，有 3 个玩家，因此有 3! 个排列. 每一个玩家在这些排列的两个排列中都是中轴. 例如，正如我们所提到的那样，玩家 2 在排列 1, 2, 3 和排列 3, 2, 1 中是中轴. 因此，每一个玩家有势力 2/3! =1/3. 在游戏[51, 49, 48, 3]中，玩家 1 在排列 2, 1, 3 和排列 3, 1, 2 中是中轴. 在第一个排列中，他带来足够多的投票把玩家 2 的投票从 48 变成 97，而在第二个排列中，他带来足够多的投票把玩家 3 的投票从 3 变成 52. 因此，

$$p_1=\frac{2}{3!}=\frac{1}{3}.$$

类似地，玩家 2 在排列 1, 2, 3 和 3, 2, 1 中是中轴，而玩家 3 在排列 1, 3, 2 和 2, 3, 1 中是中轴. 因此，

$$p_2=p_3=\frac{1}{3}.$$

因此，如预测的那样，较小的第三政党与两个较大的政党的势力相同.

在游戏[51; 40, 30, 15, 15]中，可能的排列如表 2.10 所示，其中每一个排列中的中轴玩家都被标上圆圈. 玩家 1 是中轴的有 12 项，所以他的势力是 12/4! =1/2. 玩家 2, 3, 4 各为中轴的有 4 项，所以他们每一个的势力是 4/4! =1/6.

表 2.10　游戏[51; 40, 30, 15, 15]中玩家的所有排列且中轴玩家被标上圆圈

1	②	3	4	2	①	3	4	3	①	2	4	4	①	2	3
1	②	4	3	2	①	4	3	3	①	4	2	4	①	3	2
1	③	2	4	2	3	①	4	3	2	①	4	4	2	①	3
1	③	4	2	2	3	④	1	3	2	④	1	4	2	③	1
1	④	2	3	2	4	①	3	3	4	①	2	4	3	①	2
1	④	3	2	2	4	③	1	3	4	②	1	4	3	②	1

计算澳大利亚政府，即游戏[5; 1, 1, 1, 1, 1, 1, 3]的 Shapley-Shubik 势力指数. 在本例以及下一个例子中，对于计算 Shapley-Shubik 势力指数来说，枚举出所有排列不是一个最实用的方法. 我们采用另一个方法. 联邦政府(玩家 7)在给定的排列中是中轴，当且仅当它是排列中的第三、第四或第五个位置. 根据对称性我们观察到，在这 7 个玩家的排列中，联邦政府正好在每 7 个排列中的 1 个排在第 i 个位置. 因此，它在每 7 个排列中的 3 个排在第三、第四或第五个位

76

置. 所以，联邦政府成为中轴的概率是 3/7，即：

$$p_7 = \frac{3}{7}.$$

我们也可以通过观察得到这一结果：联邦政府在 7 个玩家的排列中排在第三个位置的排列数是 6!，因为我们必须排序其余的玩家. 类似地，联邦政府排在第四个位置(或第五个位置)的排列数也是 6!. 所以，

$$p_7 = \frac{3 \cdot 6!}{7!} = \frac{3}{7}.$$

现在，很容易看到

$$p_1 + p_2 + \cdots + p_7 = 1.$$

总有(为什么)

$$\sum_i p_i = 1.$$

因此，

$$p_1 + p_2 + p_3 + p_4 + p_5 + p_6 = 1 - \frac{3}{7} = \frac{4}{7}.$$

因为根据对称性有

$$p_1 = p_2 = \cdots = p_6,$$

77

上式中的每一个数都等于

$$\frac{4}{7} \div 6 = \frac{2}{21}.$$

所以，尽管联邦政府的投票数只是一个州的投票数的 3 倍，但它的势力是州的 4½倍(9/21 比 2/21).

2.15.3 联合国安理会

接下来讨论联合国安理会的 Shapley-Shubik 势力指数的问题. 固定非常任玩家 i. 玩家 i 在在下面的排列中是中轴，即在玩家 i 的前面有所有的常任玩家及 3 个非常任玩家. 这样的排列有多少个呢? 为了找到一个这样的排列，我们首先选取在 i 前面的 3 个非常任玩家；对于每一个这样的选择，我们排序所有排在 i 前面的 8 个玩家(5 个常任玩家和 3 个非常任玩家)；对于每一个这样的选择和排序，我们排序其余 i 后面的 6 个非常任玩家. 做出第一种选择的方法数量是 $C(9, 3)$，排序前面的玩家的方法数量是 8!，且排序后面的玩家的方法数量是 6!. 因此，根据乘法规则，i 是中轴的排列数目由下式给出：

$$C(9,3) \times 8! \times 6! = \frac{9!}{3!6!} \times 8! \times 6! = \frac{9!8!}{3!}.$$

因为 15 个玩家的所有排列数目是 15!，所以

$$p_i = \frac{9!8!}{3!} \div 15! \approx 0.001865.$$

这说明非常任玩家的势力之和是这个数的 10 倍；即它是 0.018 65. 所以，因为所有势力之和是 1，常任玩家的势力之和是 0.981 35. 有五位常任玩家，根据对称性，每一个都有相同的势力. 因此它们每一个有如下势力：

$$p_j = \frac{0.98135}{5} = 0.1963.$$

因此，常任成员比非常任成员势力大 100 倍(联合国安理会和其他这种形式的立法团体的势力计算概念是由 Shapley 和 Shubik[1954]提出的).

2.15.4 两院制立法机构

为了给出一个更复杂的例子，假设我们有一个在第一院里有 n_1 个成员，在第二院里有 n_2 个成员的两院制立法机构⊖. 假设一项措施仅当它在每个立法院里都有多数赞成时能够通过，而且为了简单起见假设 n_1 和 n_2 都是奇数. 设 I 是两院的成员集合的并集，设 π 是 I 的任意排列. 在第 j 院里的玩家 i 在 π 中是中轴，如果在 π 中他是他所在院里的第 $[(n_j+1)/2]$ 个玩家，且另一院里的多半排在他的前面. 然而，对于在第一院里的玩家为中轴的任意排列 π，相反的排列使第二院里的一个玩家成为中轴(为什么). 更进一步，每一个排列以某个玩家为中轴. 所以，在院 1 的某个玩家是中轴的排列占所有排列的 1/2. 因为在院 1 里的所有玩家被一视同仁，所以这些玩家中的任意一个刚好在排列的 $1/(2n_1)$ 中成为中轴. 类似地，院 2 的任意一个玩家在排列的 $1/(2n_2)$ 中成为中轴. 因此，院 1 的每一个玩家有势力 $1/(2n_1)$ 而院 2 的每一个玩家有势力 $1/(2n_2)$. 在美国众议院和参议院中，$n_1=435$ 和 $n_2=101$，包括副总统，在平局时他投票. 根据我们的计算，每名众议员有势力 $1/870\approx0.0011$，且每名参议员(包括副总统)有势力 $1/202\approx0.005$. 因此，每名参议员的势力是一名众议员的五倍.

接下来，让我们增加一名政府官员(州长，总统)，他可以在两院里投否决票，而且假设不存在推翻否决权的可能. 现在，有 n_1+n_2+1 名玩家，一个联合是赢的，当且仅当它包含这名官员且对每个院都是多数. 假设 n_1 和 n_2 很多，Shapley 和 Shubik[1954]证明这名官员在接近一半的排列中都将是中轴(这一论证有些复杂，我们将跳过它). 这两院几乎平分余下的势力. 最后，如果加上使用两院的三分之二多数可以推翻上述的否决权，那么类似的讨论表明这名官员拥有的势力近似于六分之一，而且两院几乎平分剩余的势力. 对于更详细的讨论，我们建议读者参看 Shapley 和 Shubik 的论文.

对于在选举团中各州所使用的相关势力也可采用类似的计算. Mann 和 Shapley[1964a，b]使用如 1961 年的选举团投票分布做了这一计算. 纽约州有总共 538 张选举团投票中的 43 张，因而有势力 0.0841. 把这个数与诸如阿拉斯加州等州的势力 0.0054 比较，阿拉斯加州有三张选举团投票. 根据 Shapley 和 Shubik 的势力指数，纽约州的势力超过了它的投票的百分比，而阿拉斯加州的势力则落后于它的投票的百分比.

Boyce 和 Cross[未出版的考查，1973]得到了 1972 年选举团投票分布的类似结果. 在 1972 年的情况中，纽约州总共得到 41 张选举团投票，(总数仍为 538)且势力是 0.0797，而阿拉斯加州仍然得到三张选举团投票，且有势力 0.0054. 对于选举团游戏中的更全面的讨论可以参见 Brams，Lucas，Straffin[1983]、Lucas[1983]、Shapley[1981]和 Straffin[1980].

2.15.5 成本分摊

诸如 Shapley-Shubik 势力指数和更一般的 Shapley 值这样的游戏理论的解决方案，长期以来被用于共享项目中不同用户的成本分摊问题. 这样的应用例子包括：把飞机跑道的费用分摊给飞机场的不同用户，把高速公路费用分摊给不同尺寸的卡车，摊派不同学院共享图书馆设施的成本以及用户之间分摊电话呼叫费用. 参见 Lucas[1981a]、Okada，Hashimoto 和 Young[1982]、Shubik[1962]、Straffin 和 Heaney[1981]、Young[1986].

最近，在多点传输，如网络电影传输等方面，这些思想找到了令人着迷的应用. 在单点传输路径选择中，从源头发出的每一个信息包传送给单一接收者. 为了把相同的信息包送给多个场所，源头需要发送出这个信息包的多个拷贝，结果造成大量的宽带浪费. 在多点传输路径选择

⊖ 这个例子也是由 Shapley 和 Shubik[1954]提出的.

中，我们使用一棵“有向树”把源头与所有接收者连接起来，而在分支点一个信息包被复制成所需的份数. 用于多点传输的宽带不直接归属于单一接收者，所以必须寻找一种方法在各个接收者之间分配费用. Feigenbaum，Papadimitriou 和 Shenker[2000]、Herzog，Shenker 和 Estrin[1997]、Jain 和 Vazirani[2001]运用 Shapley 值确定了在多点传输中的成本分摊，而且研究了实现他们的方法的计算难度.

2.15.6 特征函数

本节主要讨论了简单游戏，这些游戏可以通过给每一个联合 S 赋一个等于 0 或 1 的值 $S(v)$ 来定义. 这个值通常被解释为一个联合通过合作可以得到的最好结果. 如果这个值函数或特征函数(characteristic function)能够取任意实数为值，那么这个游戏被称为特征函数形式(characteristic function form)的游戏. 最近几年这种游戏已找到了广泛的应用，例如，在水污染以及空气污染、裁军和商贸等方面. 对于这些应用的概括，可以参见 Brams，Schotter 和 Schwödianer[1979]或 Lucas[1981a]. 对于更多关于特征函数形式的游戏理论的信息，可以参见 Fudenberg，Tirole[1991]、Myerson[1997]、Owen[1995]或 Roberts[1976].

关于更多的游戏理论可以参见 Aumann 和 Hart[1998]、Fudenberg 和 Tirole[1991]、Jones[1980]、Lucas[1981b]、Myerson[1997]、Owen[1995]或 Stahl[1998].

本节练习

1. 对于下面每一个加权多数游戏，陈述所有赢联合.
   (a)[65；50，30，20]　　(b)[125；160，110，10]
   (c)1964 年的纽约纳苏郡的郡议会：[59：31，31，21，28，2，2]
   (d)[80；44，43，42，41，5]　　(e)[50；35，35，35，1]
2. 对于下列加权多数游戏，识别所有最小赢联合，即具有如下性质的赢联合：任何玩家的退出都会导致输联合.
   (a)[14；6，6，8，12，2]　　(b)[60；58，7，1，1，1，1]
   (c)[20；6，6，6，6]　　(d)练习 1 中的所有游戏

80

3. 为下面加权多数游戏中的每一个玩家计算 Shapley-Shubik 势力指数.
   (a)[51；49，47，4]　　(b)[201；100，100，100，100，1]
   (c)[151；100，100，100，1]　　(d)[51；26，26，26，22]
   (e)[20；8，8，4，2][提示：玩家 4 能成为中轴吗?]
4. 为下面游戏计算 Shapley-Shubik 势力指数.
   (a)[16；9，9，7，3，1，1](这一游戏是 1958 年的纽约纳苏郡的郡议会，参见 Banzhaf[1965]).
   (b)[59；31，31，21，28，2，2](这一游戏是 1964 年的纽约纳苏郡的郡议会，参见 Banzhaf[1965]).
5. 考虑由三个参议员 x，y，z 和三个众议院成员 a，b，c 组成的会议委员会. 一项措施在这一委员会通过，当且仅当它得到至少两名参议员和至少两名众议院成员的支持.
   (a)确定这一简单游戏的赢联合.
   (b)证明这一游戏不是加权多数游戏. 即我们无法找到投票 $v(x)$，$v(y)$，$v(z)$，$v(a)$，$v(b)$，$v(c)$和一个限额 q，使得一项措施通过当且仅当赞成它的投票总和至少是 q.（注意：一个类似的论点主张，一般不可以认为两院制立法机构是加权多数游戏.）
6. 在带有相同赢联合的加权多数游戏存在的意义下，下面的陈述中哪个定义加权多数游戏？证明你的答案.
   (a)三个玩家，一个联合能赢当且仅当玩家 1 在其中.
   (b)四个玩家 a，b，x，y，一个联合能赢当且仅当至少 a 或 b 和至少 x 或 y 在其中.
   (c)四个玩家，一个联合能赢当且仅当至少三个玩家在其中.

7. 假设一个国家有三个省. 在一个州立法机构中每个省的代表数量如下给定：省 A 有 6 名，省 B 有 7 名，省 C 有 2 名. 如果一个省的所有投票相同，且需要三分之二多数的投票，使用 Shapley-Shubik 势力指数求每个省的势力.
8. 计算会议委员会(练习 5)的 Shapley-Shubik 势力指数.
9. 证明在两院制立法中，对于每一个中轴是第一院的玩家的任意排列，其反向排列的中轴是第二院的玩家.
10. (Lucas[1983])在原来的联合国安理会中，有 5 个常任成员和 6 个非常任成员. 赢联合由所有 5 个常任成员和至少 2 个非常任成员组成.

81

(a)作为加权多数游戏公式化这一游戏.

(b)计算 Shapley-Shubik 势力指数.

11. (Lucas[1983])有人提议把日本作为第六个常任成员加入到安理会中. 如果是这样，假设仍然有 10 个非常任成员，且赢联合由所有六个常任成员和至少四个非常任成员组成.

(a)作为加权多数游戏公式化这一游戏.

(b)计算 Shapley-Shubik 势力指数.

12. 一个游戏中，有 6 个玩家每人有 11 张投票，有 12 个玩家每人有 1 张投票，要赢需要 71 张投票，计算该游戏中有一张投票的玩家的 Shapley-Shubik 势力指数.
13. 如果不需要输联合的每个子集都是输联合，或者不需要对于所有 S，或者 S 或者 $1-S$ 是输联合，那么在一个有 n 个玩家的集合上，有多少种不同的简单游戏?
14. 在一个简单游戏中，如果 p_i 是玩家 i 的 Shapley-Shubik 势力指数，且 $\sum_{i\in S} p_i$ 大于 $\frac{1}{2}$，S 一定是赢联合吗? 为什么?
15. 假设如果 S 是赢联合则 $v(S)$ 是 1，如果 S 是输联合则 $v(S)$ 是 0. 如果 p_i 是玩家 i 的 Shapley-Shubik 势力指数，证明:

$$p_i=\sum\{\gamma(s)[v(S)-v(S-\{i\})]：S \text{ 使得 } i\in S\}, \tag{2.9}$$

其中

$$s=|S| \text{ 且 } \gamma(s)=\frac{(s-1)!(n-s)!}{n!}.$$

16. 运用练习 15 中的公式(2.9)，计算练习 3 中每个加权多数游戏的 Shapley-Shubik 势力指数.
17. 通常假设如果 v 是一个特征函数，那么

$$v(\varnothing)=0, \tag{2.10}$$

$$v(S\cup T)\geqslant v(S)+v(T) \text{ 当 } S\cap T=\varnothing \text{ 时}. \tag{2.11}$$

下面 $I=\{1, 2, \cdots, n\}$ 上的特征函数中哪些有这两个性质?

(a) $n=3$，$v(\varnothing)=0$，$v(\{1\})=v(\{2\})=v(\{3\})=-1$，$v(\{1, 2\})=3$，$v(\{1, 3\})=3$，$v(\{2, 3\})=4$，$v(\{1, 2, 3\})=2$.

(b) $n=3$，$v(\varnothing)=0$，$v(\{1\})=v(\{2\})=0$，$v(\{3\})=-1$，$v(\{1, 2\})=1$，$v(\{1, 3\})=0$，$v(\{2, 3\})=-1$，$v(\{1, 2, 3\})=0$.

82

(c) n 任意；对所有 S，$v(S)=-|S|$.

18. 证明下列 $I=\{1, 2, 3, 4\}$ 上的特征函数满足练习 17 的条件(2.10)和(2.11).[⊖]

$$v(\varnothing)=0,$$

$$v(\{i\})=0 \qquad \text{所有 } i,$$

$$v(\{i,j\})=\frac{i+j}{10} \qquad \text{所有 } i\neq j,$$

$$v(\{i,j,k\})=\frac{i+j+k}{10} \qquad \text{所有不同的 } i,j,k,$$

⊖ 此练习和下一个练习是由 A. W. Tucker 提出的未发表的练习.

$$v(\{1,2,3,4\}) = \frac{1+2+3+4}{10}.$$

19. 扩展练习 18，设 $I=\{1, 2, \cdots, 2n\}$且设

$$v(\varnothing) = 0,$$
$$v(\{i\}) = 0 \quad 对所有\ i,$$
$$v(S) = \sum_{i\in S} c_i \quad 对每一个满足\ |S|>1\ 的\ S,$$

其中 c_i 是 $2n$ 个正常数且其和等于 1. 证明 v 满足练习 17 的条件(2.10)和(2.11).

20. 在一个被称为*障碍物*(deterrence)的游戏中，n 个玩家中的每一个玩家都有破坏其他玩家财产的工具. 如果 w_i 是玩家 i 的财产，那么 $v(S)$由下式给出：

$$v(S) = \begin{cases} -\sum_{i\in S} w_i & 如果\ |S|<n \\ 0 & 如果\ |S|=n. \end{cases}$$

证明 $v(S)$满足练习 17 的条件(2.10)和(2.11).

21. 在一个被称为*纯交易*(pure bargaining)的游戏中，一个私人基金向 n 个州总共提供 d 美元来开发水污染清除设备，条件是各州同意这笔资金的分配方案. 在这一游戏中 $v(S)$由下式给出：

$$v(S) = \begin{cases} 0 & 如果\ |S|<n \\ d & 如果\ |S|=n. \end{cases}$$

证明 $v(S)$满足练习 17 中的条件(2.10)和(2.11).

22. 在一个被称为*双买主市场*(two-buyer market)的游戏中，玩家 1 拥有一个对他来说价值为 a 个单位的物品. 玩家 2 认为这一物品值 b 个单位，且玩家 3 认为它值 c 个单位. 假设 $a<b<c$，这一游戏的特征函数由下式给出：

$$v(\{1\}) = a \qquad v(\{2\}) = 0 \qquad v(\{3\}) = 0 \qquad v(\{1,2\}) = b$$
$$v(\{1,3\}) = c \qquad v(\{2,3\}) = 0 \qquad v(\{1,2,3\}) = c.$$

证明 $v(S)$满足练习 17 的条件(2.10)和(2.11).

83

23. (Shapley 和 Shubik[1969])在*垃圾游戏*(garbage game)中，n 个玩家中的每个人有一袋必须扔在其他人院里的垃圾. b 袋垃圾的价值或效用是$-b$. 此时，

$$v(S) = \begin{cases} 0 & 如果\ s=0 \\ -(n-s) & 如果\ 0<s<n \\ -n & 如果\ s=n, \end{cases}$$

其中 $s=|S|$. 证明 $v(S)$满足练习 17 的条件(2.10)和(2.11).

24. 练习 15 中的公式(2.9)可以作为特征函数形式游戏的 Shapley-Shubik 势力指数的定义. 对下面各题使用这一公式：
(a)计算练习 20 的游戏的 Shapley-Shubik 势力指数.
(b)计算练习 21 的游戏的 Shapley-Shubik 势力指数.
(c)计算练习 22 的游戏的 Shapley-Shubik 势力指数，假设 $a=3$，$b=5$，$c=10$.
(d)计算练习 23 的游戏的 Shapley-Shubik 势力指数.
(e)计算练习 19 的游戏的 Shapley-Shubik 势力指数.

2.16　生成排列和组合[⊖]

在例 2.10 和例 2.17 中，我们通过检查一个集合的每一种可能排列讨论了用到的算法.（有时我们也许对每一个 r 组合或每一个子集感兴趣）在例题中我们没有提及如何确定检查排列的顺

⊖ 本节可以跳过.

序问题，因为我们认为这样的算法通常不是非常有效. 然而，有时这样的算法有用. 与此相关的是，我们需要一个生成集合的所有排列的过程，而且一般地，生成组合对象特定类型中的所有成员. 本节我们讨论这样的过程.

2.16.1　生成排列的算法

检查排列的一个自然顺序是字典序(lexicographic order). 为了描述这一顺序，假设$\pi=\pi_1\pi_2\pi_3$和$\sigma=\sigma_1\sigma_2\sigma_3$是集合{1，2，3}的两个排列. 我们说$\pi$超前于(precede)$\sigma$，如果$\pi_1<\sigma_1$，或$\pi_1=\sigma_1$且$\pi_2<\sigma_2$.(注意如果$\pi_1=\sigma_1$和$\pi_2=\sigma_2$，则$\pi_3=\sigma_3$.)例如，$\pi=123$超前于$\sigma=231$，因为$\pi_1=1<2=\sigma_1$，而$\pi=123$超前于$\sigma=132$，因为$\pi_1=1=\sigma_1$且$\pi_2=2<3=\sigma_2$. 更一般地，假设$\pi=\pi_1\pi_2\cdots\pi_n$和$\sigma=\sigma_1\sigma_2\cdots\sigma_n$是集合{1，2，…，$n$}的两个排列. 那么$\pi$超前于$\sigma$，如果$\pi_1<\sigma_1$，或$\pi_1=\sigma_1$且$\pi_2<\sigma_2$，或$\pi_1=\sigma_1$且$\pi_2=\sigma_2$且$\pi_3<\sigma_3$……或$\pi_1=\sigma_1$且$\pi_2=\sigma_2$且……且$\pi_k=\sigma_k$且$\pi_{k+1}<\sigma_{k+1}$，或……因此，$\pi=42135$超前$\sigma=42153$，因为4=4，2=2，1=1，且3<5. 使用这种字典序，如同在字典中排列单词一样排序，即先考虑第一个"字母"，在相同的情况下考虑第二个"字母"，以此类推. 下面以字典序列出集合{1，2，3}的所有排列：

84

$$123,\ 132,\ 213,\ 231,\ 312,\ 321. \tag{2.12}$$

注意就数字集合{1，2，3，…，n}的排列而言，如果$n\leqslant 9$，那么这个集合的排列本身可以看成是"数". 例如，排列3241可以看成是三千二百四十一. 在这种情况下，排列的字典序等价于这个"数字"的升序. 在(2.12)式中，以字典序的排列从数字一百二十三递增到三百二十一.

我们将描述以字典序列出所有排列的算法. 其关键步骤是，对于给定的排列$\pi=\pi_1\pi_2\cdots\pi_n$确定下一个排列. 以字典序的最后一个排列是$n$，$n-1$，$n-2$，…，2，1. 以该顺序它没有下一个排列. 所有其他排列π对于某个i有$\pi_i<\pi_{i+1}$. 如果$\pi_{n-1}<\pi_n$，这一顺序的下一个排列可以通过交换π_{n-1}和π_n而得到. 例如，如果$\pi=43512$，那么下一个排列是43521. 现在假设$\pi_{n-1}>\pi_n$. 如果$\pi_{n-2}<\pi_{n-1}$，我们可以重排列π的最后三项以得到这一顺序下的下一个排列. 特别地，我们考虑π_{n-1}和π_n且寻找比π_{n-2}大的数字中最小的那一个. 我们把这个数字放在第$(n-2)$个位置上. 然后，以递增顺序排列最后三个数字中剩余的两个数字. 例如，假设$\pi=15243$，那么$\pi_{n-1}=4>3=\pi_n$，但是$\pi_{n-2}=2<4=\pi_{n-1}$. π_{n-1}和π_n都大于π_{n-2}且3是π_{n-1}和π_n中较小的一个. 因此，把3放置到第三个位置，并把2和4按递增顺序放置，从而得到15324. 如果$\pi=15342$，我们将把4切换到第三个位置，而不是2，因为2<3，从而得到15342.

一般地，如果$\pi\neq n$，$n-1$，$n-2$，…，2，1，那么一定存在使得$\pi_i<\pi_{i+1}$的最右侧的i. 那么必须重排列从π_i开始的元素以便找到这一顺序下的下一个排列. 通过检查所有满足$j>i$的π_j，并寻找大于π_i的最小π_j来实现这一点. 然后，互换π_i和π_j. 互换后，按递增顺序放置π_j后面的数字. 它们现在是递减顺序，所以简单地翻转它们就可以了. 例如，假设$\pi=412653$. 那么$\pi_i=2$和$\pi_j=3$. 交换π_i和π_j得到413652. 然后翻转得到413256，这就是字典序下的下一个排列.

这一算法的各步骤概括如下：

算法 2.1：生成 1，2，…，n 的所有排列

输入： n.

输出： {1，2，…，n}字典序下的所有$n!$个排列的列表.

步骤 1. 设$\pi=12\cdots n$且输出π.

85

步骤 2. 如果对于所有的i，$\pi_i>\pi_{i+1}$，则停止.（上述列表完成.）

步骤 3. 寻找使得$\pi_i<\pi_{i+1}$的最大i.

步骤 4. 寻找使得 $i<j$ 且 $\pi_i<\pi_j$ 的最小 π_j.

步骤 5. 交换 π_i 和 π_j.

步骤 6. 以新的顺序翻转 π_j 后面的数字，设 π 为这一结果的排列，输出 π 并返回到步骤 2.

注意，可以修改算法 2.1，使其在生成排列时做各种目的的检查. 关于算法 2.1 的计算机实现的细节参见 Reingold，Nievergelt 和 Deo[1977].

2.16.2 生成集合子集的算法

在 2.6 节，我们考虑了对于给定的浇头集合寻找所有可能的比萨的问题. 这等价于寻找给定的集合的所有子集的问题. 本节论述解决这一问题的算法.

从假设 S 是集合$\{1, 2, \cdots, n\}$的一个子集开始. 表示 S 的一个等价的方法是，使用长度为 n 的位串 B，其中 1 在 B 的第 i 位上表明 i 在 S 中，而 0 在 B 的第 i 位上则表明 i 不在 S 中. 例如，如果 $S=\{1, 3, 4, 6\}\subseteq\{1, 2, 3, 4, 5, 6, 7\}$，则 $B=1011010$. 因此，生成 n 集合的所有子集的问题变成生成长度为 n 的所有位串的问题.

对这些位串将使用的排序类似于为排列引入的字典序. 假设 $\alpha=\alpha_1\alpha_2\alpha_3$ 和 $\beta=\beta_1\beta_2\beta_3$ 是长度为 3 的两个位串. 我们说 α 超前于 β 如果 $\alpha_1<\beta_1$，或如果 $\alpha_1=\beta_1$ 且 $\alpha_2<\beta_2$，或如果 $\alpha_1=\beta_1$ 且 $\alpha_2=\beta_2$ 且 $\alpha_3<\beta_3$. 例如，$\alpha=001$ 超前于 $\beta=010$，因为 $\alpha_1=0=\beta_1$，且 $\alpha_2=0<1=\beta_2$. 更一般地(和更简洁地)，假设 $\alpha=\alpha_1\alpha_2\cdots\alpha_n$ 和 $\beta=\beta_1\beta_2\cdots\beta_n$ 是两个长度为 n 的位串. 那么 α 超前 β 如果对于 α_i 不同于 β_i 的最小的 i，有 $\alpha_i<\beta_i$. 因此，$\alpha=01010$ 超前 $\beta=01100$，因为 $\alpha_1=0=\beta_1$，$\alpha_2=1=\beta_2$ 且 $\alpha_3=0<1=\beta_3$. 在这一"字典"顺序下，我们再一次如同在字典中排列单词那样排序；然而，此时，我们研究的是受限的字母表，0 和 1 的情况. 下面以字典序列出所有长度为 3 的位串：

$$000, 001, 010, 011, 100, 101, 110, 111. \tag{2.13}$$

像数字那样处理位串并使用递增顺序是考虑一个集合子集的字典排序的另外一种方法. 注意，在(2.13)式中，长度为 3 的位串从 0 开始一直到一百一十一，且在中间是递增的.

接下来，我们给出以字典序列表所有位串的一个算法. 给定一个位串，这个位串的下一个位串是什么呢? 因为 11…1 没有 0，所以根据上述超前的定义，它不会超前任意其他位串，且它将是这一顺序下的最后一个位串. 对于任意其他位串 β，存在使 $\beta_i=0$ 的 i. 在这一顺序下 β 之后的下一个位串可以通过下面的方法得到：开始于 β_n 且向后处理，把所有遇到的 1 都变成 0，反之亦然. 只要在第一次把 0 变成 1 后就停止这一过程，可以得到这一顺序下的下一个位串. 例如，假设 $\beta=1001011$. 我们改变 β_5，β_6 和 β_7 的位置(下面用黑体表示)，从而得到下面的位串：[86]

$$100\mathbf{1011}\rightarrow 100\mathbf{1100}$$

另外，如果我们把位串看成是"数"，那么 β 之后的下一个位串是下一个最大位串. 这可以通过把 1 加入到 β 而得到. 不难看出把 1 加入到 β 实现的效果与我们上面所描述的一样.

这一算法的步骤概括如下：

算法 2.2：生成长度为 n 的所有位串

输入： n.

输出： 以字典序排列的长度为 n 的所有 2^n 个位串的列表.

步骤 1. 设 $\beta=00\cdots0$ 且输出 β.

步骤 2. 如果对于所有的 i，$\beta_i=1$，停止.（列表完成）

步骤 3. 寻找使得 $\beta_i=0$ 的最大 i.

步骤 4. 把 β_i 变成 1，且 β_{i+1}，β_{i+2}，…，β_n 变成 0，设 β 为这一结果的位串，输出 β 且返回步骤 2.

对于长度为 n 的位串(及 $\{1, 2, \cdots, n\}$ 的排列)，一定存在字典类型顺序之外的其他顺序. 我们描述长度为 n 的所有位串的另外一个顺序. 用来检查这些位串的新顺序称为**二进制反射格雷码顺序**(binary-reflected Gray code order)㊀(当我们在下面描述这一顺序时，术语**二进制反射**就会变得清楚. **代码**一词的使用源于它与代码理论的关系，参见第 10 章).

容易递归地定义长度为 n 的位串的二进制反射格雷码顺序，记为 $G(n)$. 即使用长度小于 n 的位串的二进制反射格雷码顺序来定义长度为 n 的位串的二制反射格雷码顺序. 使用 $G_i(n)$ 表示顺序 $G(n)$ 下的第 i 个位串. 一般地，二进制反射格雷码顺序开始于所有位为 0 的位串，回想一下 n 集合的子集的数量是 2^n. 因此，$G(1)$ 是开始于位串 $0=G_1(1)$ 且结束于位串 $1=G_2(1)$ 的顺序，即

$$G(1) = 0,1.$$

为了求 $G(2)$，我们列出 $G(1)$ 的元素，且在每一个元素的开始位置上附加一个 0. 接着以"相反的顺序"列出 $G(1)$ 的元素，而且在这些元素的每一个的开始位置上附加一个 1. 因此有

87

$$G(2) = 0G_1(1), 0G_2(1), 1G_2(1), 1G_1(1) = 00, 01, 11, 10. \tag{2.14}$$

无论是从 $G(1)$ 到 $G(2)$ 还是从 $G(n)$ 到 $G(n+1)$，都使用相同的方法.

$$G(n+1) = 0G_1(n), 0G_2(n), \cdots, 0G_{2^n}(n), 1G_{2^n}(n), 1G_{2^n-1}(n), \cdots, 1G_1(n). \tag{2.15}$$

设 $G(n)^R$ 是 $G(n)$ 的逆序. 使用一个不太严格的记法，$G(n+1)$ 可以定义为

$$G(n+1) = 0G(n), 1G(n)^R. \tag{2.16}$$

因此，

$$G(3) = 0G(2), 1G(2)^R = 000, 001, 011, 010, 110, 111, 101, 100.$$

注意，从 $G(1)$ 到 $G(2)$ 及从 $G(2)$ 到 $G(3)$ 时，元素的数量成倍增加. 这不是反常现象. $|G(2)| = 2^2 = 2 \cdot 2^1 = 2|G(1)|$ 且一般地

$$|G(n+1)| = 2^{n+1} = 2 \cdot 2^n = 2|G(n)|.$$

等式(2.15)定义的 $G(n+1)$ 有 $2 \cdot 2^n = 2^{n+1}$ 个元素，且没有重复元素. (为什么?)因此，递归定义的二进制反射格雷码顺序 $G(i)$，事实上是长度为 i 的所有位串的一个排序，其中 $i=1, 2, \cdots$. 与递归定义相反，可以寻找直接列出 $G(n)$ 所有项的算法. 我们把这一工作留给读者(练习 15)完成. 它的关键步骤是，给定长度为 n 的位串 $\beta=\beta_1\beta_2\cdots\beta_n$，生成它在 $G(n)$ 中的下一个位串.

尽管这一问题的字典序可能更加直观，但是，二进制反射格雷码顺序在另一个意义下更好些. 有时候在某一顺序下把两个连续元素间的变动保持在最小是很重要的. 在这方面，二进制反射格雷码顺序一定比字典序更有效. 二进制反射格雷码顺序的连续元素仅在一处不同. 这显然是最好的. 长度为 n 的位串的字典序对于某一对连续元素总是有 n 处变动(为什么). 注意，在(2.13)式中，从 011 到 100 时存在 $n=3$ 个变动处.

2.16.3　生成组合的算法

继排列和子集之后本章定义的下一个组合数学对象是 n 集合的 r 组合. 使用前一节的集合的子集术语，这些组合可以认为是大小正好为 r 的集合 $\{1, 2, \cdots, n\}$ 的子集或正好带有 r 个 1 的长度为 n 的位串. 因为有这样的关系，生成它们的算法与算法 2.2 紧密相关.

假设我们感兴趣的是 5 集合的 3 组合的生成. 因为我们考虑的是子集，所以与顺序无关. 于是，

$$\{1,2,5\}, \{1,5,2\}, \{2,1,5\}, \{2,5,1\}, \{5,1,2\}, \{5,2,1\}$$

都是相同的. 因此，作为一种选择，我们的算法将生成每个子集的元素为升序的 r 组合. 所以，

88

对于上面列出的 6 个相同子集，我们的算法将生成 $\{1, 2, 5\}$. 有 $\binom{5}{3}=10$ 个 5 集合的 3 组合，

㊀ 这一顺序是基于格雷的工作[1953].

按字典序递增排列为

$$\{1,2,3\},\{1,2,4\},\{1,2,5\},\{1,3,4\},\{1,3,5\},\{1,4,5\},\{2,3,4\},\{2,3,5\},\{2,4,5\},\{3,4,5\}.$$

n 集合的 r 组合生成算法的工作方式如下：给定一个 r 组合，$\gamma=\{\gamma_1, \gamma_2, \cdots, \gamma_r\}$，寻找最大 i 使得$\{\gamma_1, \gamma_2, \cdots, \gamma_{i-1}, \gamma_{i+1}, \gamma_{i+1}, \cdots, \gamma_r\}$是一个 r 组合且它的元素仍是升序. 然后，重新设置 $\gamma_{i+1}, \gamma_{i+2}, \cdots, \gamma_r$ 为它们的最低可能值. 例如，假设 $\gamma=\{1, 3, 4, 7, 8\}$是 8 集合$\{1, 2, 3, 4, 5, 6, 7, 8\}$的一个 5 组合，它的元素处于递增状态. 于是 $i=3$ 且 $\gamma_i=4$. 给 γ_i 加 1 得到$\{1, 3, 5, 7, 8\}$. 可以把 γ_4 和 γ_5 分别重新设置成 6 和 7，给出结果$\{1, 3, 5, 6, 7\}$，这一组合是递增字典序下的下一个 5 组合. 这一算法开始于$\{1, 2, \cdots, r\}$，结束于$\{n-r+1, n-r+2, \cdots, n\}$，因为其后没有元素可以增加.

这一算法，如同我们前面所述的字典算法一样，在最小化变动这一顺序下的两个相继元素的意义下不是有效的. 类似于前节所述算法的二进制反射格雷码顺序，在这方面最有效. 回想一下，$G(n+1)$是根据(2.16)式递归定义的. 下面设 $G(n, r)(0\leqslant r\leqslant n)$是长度为 n 且正好带有 r 个 1 的位串的二进制反射格雷码顺序. $G(n, r)(0<r<n)$可以通过下式递归定义：

$$G(n,r) = 0G(n-1,r),1G(n-1,r-1)^R$$

和

$$G(n,0) = 0^n = \underbrace{00\cdots0}_{n\text{个}} \quad \text{且 } G(n,n) = 1^n = \underbrace{11\cdots1}_{n\text{个}}.$$

其他情况下，在 $G(n, r)$中第一个位串和最后一个位串分别是

$$0^{n-r}1^r \text{ 和 } 10^{n-r}1^{r-1},$$

例如，

$G(1, 0)=0$	$G(2, 0)=00$
$G(1, 1)=1$	$G(2, 1)=01, 10$
	$G(2, 2)=11$
$G(3, 0)=000$	$G(4, 0)=0000$
$G(3, 1)=001, 010, 100$	$G(4, 1)=0001, 0010, 0100, 1000$
$G(3, 2)=011, 110, 101$	$G(4, 2)=0011, 0110, 0101, 1100, 1010, 1001$
$G(3, 3)=111$	$G(4, 3)=0111, 1101, 1110, 1011$
	$G(4, 4)=1111.$

因为每一个位串必须正好包含 r 个 1，一个顺序下的两个相继项的最佳期待是它们至多有两位不同. 长度为 n 且正好带有 r 个 1 的位串的二进制反射格雷码顺序事实上达到了这一最佳期待.　89

有关排列、子集和组合的更多生成算法以及合成和划分的内容可以参见 Reingold, Nievergelt 和 Deo[1977]. 关于生成排列和组合的早期但全面的论文当属 Lehmer[1964].

本节练习

1. 对于下面的几对排列，按字典序确定哪一个应排在前面.
   (a)3412 和 2143　　(b)3124 和 3214
   (c)234651 和 235164　　(d)76813254 和 76813524
2. 对于下面的几对位串，按字典序确定哪一个应排在前面.
   (a)1001 和 1101　　(b)0101 和 0110
   (c)11011 和 11101　　(d)110110101 和 110101100
3. 对于下面的几对组合，按递增的字典序确定哪一个应排在前面.
   (a)$\{1, 2, 3, 4\}$和$\{1, 2, 3, 5\}$　　(b)$\{1, 2, 6, 7, 9, 16\}$和$\{1, 2, 6, 7, 10, 13\}$

(c){2, 4, 5, 6}和{1, 3, 5, 6}　(d){8, 10, 11, 12, 13}和{8, 9, 10, 11, 13}

4. 对于下面的每一个排列，寻找在字典序下跟随其后的排列.

(a)123456　(b)3457621　(c)152463

(d)82617543　(e)4567321　(f)54328716

5. 对于下面的每一个位串，寻找在二进制反射格雷码顺序下跟随其后的位串.

(a)011010　(b)0101001　(c)0000

(d)11111　(e)1110111　(f)011111

6. 对于下面的每一个 8 集合的 r 组合，寻找在递增字典序下跟随其后的组合.

(a){1, 3, 5, 6, 8}　(b){2, 3, 4, 7}

(c){1, 3, 4, 5, 6, 7}　(d){4, 5, 6, 7, 8}

7. 按字典序列出{1, 2, 3, 4}的所有排列.

8. 按字典序列出{u, v, x, y}的所有排列.

9. 按字典序列出长度为 4 的所有位串.

10. 考虑 JAVA(练习 2.8)中长度为 3 的所有变量. 如果字母排在下划线之前，下划线排在美元符号之前，所有这些排在十进制数字之前，寻找字典序下的第一个和最后一个 3 字符变量.

11. 按递增字典序，列出{1, 2, 3, 4, 5}的所有 2 组合.

12. 寻找一个长度为 n 的位串，当在字典序下生成其下一个位串时，它的所有 n 个点都发生变动.

[90]

13. 求 $G(4)$.

14. 使用等式(2.14)和(2.15)中的递归定义，证明在这一顺序下 $G(n)$没有重复项.

15. 寻找在二进制反射格雷码顺序下生成所有长度为 n 的位串的算法.

16. 使用算法 2.2，证明生成给定 n 集合的所有子集的字典序对某一对连续项总有 n 处变动.

17. 开发生成集合{1, 2, …, n}的满足如下条件的所有排列的算法：对于所有的 i，i 不出现于任意排列的第 i 个点上.

18. 根据二进制反射格雷码顺序，解释使用词汇"二进制反射"的理由. [提示：参考由等式(2.15)给出的寻找 $G(n)$的过程.]

19. 证明二进制反射格雷码顺序下的任意两个连续位串正好在一个位置上不同.

20. 对于 $0\leqslant r\leqslant 5$，求 $G(5, r)$.

21. 当 π 是集合{1, 2, …, n}的所有排列在字典序下的第 i 个排列时，令 $f_n(\pi)$是 i. 计算：

(a)$f_2(21)$　(b)$f_3(231)$

(c)$f_5(15243)$　(d)$f_6(654321)$

22. 对于练习 21 所定义的 $f_n(\pi)$，假设排列 π'由排列 $\pi=\pi_1\pi_2\cdots\pi_n$ 通过下面的方法得到的：删除 π_1 且把所有满足 $\pi_j>\pi_1$ 的 π_j 减少 1. 证明 $f_n(\pi)=(\pi_1-1)(n-1)!+f_{n-1}(\pi')$,

23. 寻找生成 n 集合的所有 r 排列的算法.

24. 使用算法 2.1 及其后的注释，寻找另一个生成 n 集合的所有 r 排列的算法.

25. 回想一下 2.4 节中的复杂度函数的概念. 计算：

(a)算法 2.1 的复杂度函数.

(b)算法 2.2 的复杂度函数.

(c)按递增字典序生成 n 集合的所有 r 排列的算法的复杂度函数.

2.17 排列间的倒位距离和突变研究[⊖]

突变是物种进化的一个关键过程. 这些突变出现在表示 DNA 的序列中. 有时候，突变可能包含*倒位*(inversion)(DNA 片段被反转)、*互换*(transposition)(两个相邻的 DNA 片段交换位置)、

⊖ 本节基本上来自 Gusfield[1997]. 跳过本节不会影响对后面内容的理解.

易位(translocation)(两个染色体的尾端交换)，有时候可能更复杂. 最新的工作集中于对这些突变的算法分析上. 例如，参见 Ferretti，Nadeau 和 Sankoff[1996]、Hannenhalli 和 Pevzner[1996]、Kaplan，Shamir 和 Tarjan[1997]. 本节集中研究倒位. [91]

倒位在某些物种中似乎是突变的主要形式. 例如，倒位在果蝇(Dobzhansky[1970])、豌豆类植物(Palmer，Osorio 和 Thompson[1988])和某些细菌(O'Brien[1993])的进化中起特殊的作用. 我们可以研究基因的倒位，其中基因对应于 DNA 的子序列. 基因被排列在染色体上，而且在给定染色体上的基因可区分的初始假设是合理的，所以可以想象它们在给定染色本上被标上标签 1，2，…，n，而且经一系列倒位后成为{1，2，…，n}的一个排列. 于是倒位操作反转这一排列的子序列. 例如，从恒等排列 123456 开始，我们可以颠倒子序列 2345 得到 154326，然后颠倒子序列 15 得到 514326，然后颠倒子序列 432 得到 512346. 在进化的研究中，我们知道当前物种或 DNA 序列或基因序列是什么，而且尝试着再现如何得到它. 于是，一个自然的问题是如何以最少步骤，或在当前情况下，以最小的倒位数，从一个排列，如 123456 得到另外一个排列，如 512346. 这一问题被称为从排列 123456 到排列 512346 的*倒位距离*(inversion distance)[㊀]或简称为 512346 的倒位距离. 我们已知道如何以三个倒位从 123456 得到 512346. 事实上，这一倒位距离至多为 2：颠倒子序列 12345 得到 543216，然后颠倒子序列 4321 得到 512346. 因为，512346 的倒位距离不能是 1，因此，它的倒位距离等于 2. (为什么)我们将描述一个启发式算法，该算法寻找从恒等排列 12…n 得到{1，2，…，n}的排列 π 所需要的最少倒位数量，即计算 π 的倒位距离.

如果 π 表示一个排列，那么 π_i 表示 π 中第 i 个位置的数字. 换句话说，$\pi=\pi_1\pi_2\cdots\pi_n$. 例如，在排列 512346 中 $\pi_3=2$. π 中的*断裂点*(breakpoint)出现在两个数字 π_i 和 π_{i+1} 之间，如果

$$|\pi_i-\pi_{i+1}|\neq 1.$$

另外，如果 $\pi_1\neq 1$，π 在它的前端有断裂点；如果 $\pi_n\neq n$，π 在它的末端有断裂点. 所以，$\pi=143265$ 在 1，4 间、2，6 间和它的末端有断裂点.

设 $\Phi(\pi)$ 等于 π 中的断裂点总数.

定理 2.10　*任意排列 π 的倒位距离至少是* $\left\lceil \frac{\Phi(\pi)}{2} \right\rceil$.

证明　如果 $\pi=\pi_1\pi_2\cdots\pi_n$ 且子序列 $\pi_i\pi_{i+1}\cdots\pi_{i+j}$ 被颠倒，那么通过这一倒位至多生成两个新断裂点. 这两个新断裂点可能出现在下面的新排列的 π_{i-1} 和 π_{i+j} 之间或/和 π_i 和 π_{i+j+1} 之间：

$$\pi^*=\pi_1\pi_2\cdots\pi_{i-1}\pi_{i+j}\pi_{i+j-1}\cdots\pi_{i+1}\pi_i\pi_{i+j+1}\pi_{i+j+2}\cdots\pi_n.$$

因为恒等排列没有断裂点，且把恒等排列转换成 π(反之亦然)至少需要 $\lceil\Phi(\pi)/2\rceil$ 个倒位. ■ [92]

上面已经注意到排列 143265 有三个断裂点. 因此，根据定理 2.10，它的倒位距离至少是 $\lceil 3/2\rceil=2$.

一个断裂点和(a)该排列的前端间，或(b)另一个断裂点，或(c)该排列的末端间的排列中的子序列没有其他断裂点时，这一子序列称为带(strip). 如果在这一条带中的数字是递增(递减)的，那么这条带称为*递增*(*递减*). 由单一数字组成的带被定义为递增[㊁]. 例如，排列 541236 有 3 个断裂点和 3 条带. 两条递减带是 54 和 6，而递增带是 123.

已经证明求解排列的倒位距离问题是一个困难的问题[㊂]. 有时候，近似求解倒位距离的方法或至少在一个固定因子之内的方法容易找到. 下面的两个引理可以用来给出把任意排列转换成

㊀ 也译成演化距离. ——译者注

㊁ 原文为递减，根据下面的两个引理，应为递增. ——译者注

㊂ 使用 2.18 节的语言，倒位距离的计算是一个 NP-难题(Caprara[1997]).

恒等排列，且使用的倒位数量至多是倒位距离的四倍的算法.

引理 2.1 如果排列 π 包含一个递减带，则存在减少断裂点数目的倒位.

证明[⊖] 考虑带有包含在任意递减带内最小数 π_i 的递减带. 根据定义，π_i 在这个带的右端. 如果 $\pi_i=1$，那么 $\pi_1\neq1$，在这种情况下，在 π_1 之前和 π_i 之后一定存在断裂点. 颠倒 $\pi_1\pi_2\cdots\pi_n$ 至少减少一个断裂点，因为 1 移入这一颠倒排列的第一个点.

假设 $\pi_i\neq1$，考虑 π_{i+1}，如果存在的话. 它不能是 π_i-1，因为否则它将在递减带内. 它也不能是 π_i+1，因为否则 π_i 将不在递减带内. 所以，在 π_i 和 π_{i+1} 之间，如果 π_{i+1} 不存在，则在 π_i 之后(即在整个排列的末端)一定存在断裂点. 根据类似的推理，紧接 π_i-1 的右端一定存在一个断裂点.

如果 π_i-1 位于 π_i 的右边，那么倒位 $\pi_{i+1}\pi_{i+2}\cdots\pi_{i-1}$. 如果 π_i-1 位于 π_i 的左边，那么倒位开始于紧接 π_{i-1} 右端的项一直到 π_i. 无论是哪种情况，我们都将倒位一个在其两端都有断裂点的子序列，并至少减少一个断裂点，因为 π_i-1 和 π_i 现在是倒位后排列中的两个连续元素. ■

例如，排列 541236 中的 54 是有最小数 4 的递减带. 定位数字 3，由引理，它一定在一个递增序列中且紧接它的右边一定是一个断裂点. 颠倒 123 得到 543216. 这一新排列有 2 个断裂点，而原来的序列有 3 个.

引理 2.2 如果排列 π 不是一个恒等序列，且不包含递减带，那么存在一个倒位，它不增加断裂点的数量但是增加一个递减带.

[93]

证明[⊜] 因为不存在递减带，所以每一个带必定是递增的. 如果 $\pi_1\neq1$ 或 $\pi_n\neq n$，那么分别颠倒从 π_1 开始的递增带或首位到 π_n 的递增带满足引理. 否则，寻找 $\pi=1$ 之后的第一个和第二个断裂点. 因为 π 不是恒等排列，这些断裂点存在. 两个断裂点之间的子序列满足这一引理. ■

考虑完成如下工作的倒位算法(inversion algorithm)：

步骤 1. 如果存在一个递减带，使用引理 2.1 的倒位. 重复直到不存在递减带.

步骤 2. 使用引理 2.2 的倒位并返回到步骤 1.

每两个倒位断裂点的数量至少减少一个. 根据定理 2.10，我们有下面的定理.

定理 2.11(Kececioglu 和 Sankoff[1994]) 倒位算法把任意排列转换成恒等排列，所使用的倒位数量最多是倒位距离的四倍.

最多使用四倍于倒位数量的界限并非人们所熟知的最佳界限. Kececioglu 和 Sankoff[1995] 通过证明下面的引理把这个误差界限减少到了一半.

引理 2.3 设 π 是带有一个递减带的排列. 如果减少 π 的断裂点数量的每一个倒位都得到没有递减带的排列，那么 π 有一个减少两个断裂点数量的倒位.

使用这一新引理，本质上，我们知道总是存在两个连续的倒位使断裂点数减少 2. 这意味着我们可以使用至多 $\Phi(\pi)$ 个倒位达到恒等排列，(根据定理 2.10)它至多是倒位距离的两倍. Bafna 和 Pevzner[1996]通过考虑一个倒位对下一个倒位的效应把 Kececioglu 和 Sankoff[1995]的界限降低到 1.75.

倒位只是突变的一个类型，但是一个重要的类型，特别是在一个染色体的有机体内是一个重要的突变类型(Sessions[1990]). 其他倒位的变体将在练习中加以描述. 互换和易位也是改变排列的有趣方式. 对于前者我们提几句. 排列 $\pi=\pi_1\pi_2\cdots\pi_n$ 的互换 $\pi_i\pi_{i+1}$ 产生排列 $\pi_1\pi_2\cdots\pi_{i-1}\pi_{i+1}\pi_i\pi_{i+2}\cdots\pi_n$. 通过这种形式互换把恒等排列变成另一个排列(反之亦然)在组合数学中得到了

⊖ 这个证明可以跳过.

⊜ 这个证明可以跳过.

很好的研究. 已经确立了做这一变换所需的互换数量. 根据众所周知的公式(例如, 参见 Jerrum [1985]), 把一个恒等排列变成{1, 2, …, n}的排列 π 所需要的互换数量由下式给出:

$$J(\pi)=|\{i,j\}:1\leqslant i<j\leqslant n \,\&\, \pi_i>\pi_j\}|. \tag{2.17}$$

94

这一公式的证明留给读者(练习 8). 为了说明这一结果, 我们注意到因为 $J(\pi)=|\{(2,3),(2,6),(3,6),(4,5),(4,6),(5,6)\}|$, 把排列 $\pi=143652$ 变成恒等排列所需的互换数量是 6. 受到关于突变研究的启发, 更一般的工作允许一个排列的整个片段的互换. 例如, 通过把片段 582 和 35 互换可以把 15823647 变成 13658247. 作为参考, 可以参阅 Bafna 和 Pevzner[1998]、Christie[1996]以及 Heath 和 Vergara[1998]. 互换也出现在 Mahadev 和 Roberts[2003]中, 他们把互换应用于通信中的信道分配以及 DNA 的物理测绘.

本节练习

1. 下面的排列中哪些排列有倒位距离 1?

   (a)4567123　　(b)12354678　　(c)54321

2. 下面的排列中哪些排列有倒位距离 2?

   (a)4567123　　(b)23456781　　(c)54321

3. 给出带有 3 个断裂点而倒位距离为 3 的排列的例子. (注意, 这是定理 2.10 中的边界可能不是最好的另一个例子.)

4. 证明一个带总是递增或递减的.

5. 考虑下面计算倒位距离的贪婪算法. 首先找到并运用把 1 放到位置 π_1 上的倒位. 其次找到并运用把 2 放到位置 π_2 上的倒位, 以此类推.

   (a)把这一贪婪算法运用到 512346.

   (b)证明这一算法至多以 $\Phi(\pi)$个倒位结束.

   (c)求使用这一贪婪算法需要 $n-1$ 个倒位的 1, 2, …, n 的排列.

   (d)(c)中的所求排列的倒位距离是多少?

6. (Kececioglu 和 Sankoff[1995])在*带符号倒位问题*(signed inversion problem)中, 排列中的每一个数都有一个符号(+或−), 每当该数包含在一个倒位的子序列中时, 符号发生变化. 例如, 从排列+1−5+4+3−2−6 开始, 我们可以倒转 543 得到+1−3−4+5−2−6. 这一带符号倒位距离问题是, 使用最少的倒位数把一个带符号排列转换成其数字都带有正号的恒等排列.

   (a)给出一系列倒位, 使得把−3−4−6+7+8+5−2−1 转换成全正恒等排列.

   (b)在一个带符号排列 π 中, *邻接*(adjacency)被定义为形如$+i\quad+(i+1)$或$-(i+1)\quad-i$ 的一对连续数. *断裂点*(breakpoint)被定义为不形成邻接的任意两个连续数之间的点. 同样, 除非第一个数是+1, 否则在 π 的前端有一个断裂点; 并且除非最后一个数是$+n$, 否则在 π 的末端有一个断裂点.

   95

   i. −3−4−6+7+8+5−2−1 有多少个断裂点?

   ii. 使用类似于定理 2.10 中的断裂点的说法描述带符号倒位距离的边界.

7. 求把下面的排列转换成恒等排列的互换的最小数目, 并对每一种情况, 确定哪一组互换达到了最小数目.

   (a)54321　　(b)15423　　(c)625143

8. 证明(2.17)式(Jerrum 公式).

9. 回想一下, 当 DNA 两个相邻片段互相交换位置时会出现*互换引发的突变*(mutation due to transposition). 寻找把下面的排列转换成恒等排列的片段互换. 这样的互换的最小次数是多少?

   (a)5123746　　(b)987123456

10. 假设一个有机体有一个环状染色体. 在这种情况下, 排列 $\pi_1, \pi_2, \cdots, \pi_n$ 等价于 $\pi_i, \pi_{i+1}, \cdots, \pi_n, \pi_1, \pi_2, \cdots, \pi_{i-1}$. 我们称这样的排列为*连接的*(connected). 寻找一系列倒位, 把下面的连接排列转换成恒等排列. 倒位的最小数目是多少?

(a)5123746　　(b)987123456

2.18 好算法[⊖]

2.18.1 渐近分析

在2.4节，我们已经观察到解决组合数学问题的一些算法不是太好．本节尝试通过好算法的概念使我们的论述更精确．正如在2.4节中指出的那样，在一个特定的机器上运行特定程序的成本将随着程序员的技术及这一机器的特性而变化．因此，在计算机科学领域，我们强调的是对解决问题的算法分析而不是特定计算机程序的分析，这也将是我们在这里强调的重点．

在分析一个算法的好坏程度时，我们尝试使用2.4节的术语评估一个复杂度函数$f(n)$．如果n相对小，那么$f(n)$通常也相对地小．大多数算法对小问题来说没有任何问题．我们主要对n相对大时的复杂度函数$f(n)$的比较感兴趣．

算法分析中的关键概念如下．假设F是有复杂度函数$f(n)$的算法，且$g(n)$是n的任意函数．我们记F或f是$O(g)$，如果存在整数r和正常量k使得对于所有的$n\geqslant r$，有$f(n)\leqslant k\cdot g(n)$．[如果$f$是$O(g)$，那么有时候我们称$g$*渐近支配*(asymptotically dominate)f．]如果f是$O(g)$，那么对于输出大小至少是r的问题，带有复杂度函数f的算法的成本将永远低于带有复杂度函数g的算法的成本的k倍．作为例子，$100n$是$O(n^2)$，因为对于$n\geqslant 100$，有$100n\leqslant n^2$．同样，$n+1/n$是$O(n)$，因为对于$n\geqslant 1$有$n+1/n\leqslant 2n$．$O(n)$的算法称为是*线性*(linear)算法，$O(n^2)$的算法称为是*二次*(quadratic)算法，对于多项式g，$O(g)$的算法称为*多项式*(polynomial)算法．计算机科学中其他重要算法类型有$O(\log n)$、$O(n\log n)$、对$c>1$，$O(c^n)$和$O(n!)$．在下面或在练习中我们将讨论这些算法．

[96]

对于$c>1$，复杂度函数是c^n的算法称为*指数*(exponential)算法．注意，每一个指数算法都是$O(c^n)$，但是每一个$O(c^n)$算法不一定是指数算法．例如，复杂度函数是n的算法对于所有的$c>1$是$O(c^n)$．这是因为对于足够大的n，$n\leqslant c^n$成立．

一般可接受的原则是，如果一个算法是多项式的，那么它是*好*(good)算法．这一思想源自Edmonds[1965]．关于好算法的讨论可以参见Garey和Johnson[1979]、Lawler[1976]或Reingold，Nievergelt和Deo[1977]．这里尝试着给出好算法的简单考证[⊜]．

因为我们只对相对大的n时的$f(n)$和$g(n)$感兴趣，所以我们在定义"f是$O(g)$"的概念中引入了常量r．但是常量k来自何处呢？考虑算法F和G，它们的复杂度函数分别是$f(n)=20n$和$g(n)=40n$．现在，显然算法F更可取，因为对于所有n有$f(n)\leqslant g(n)$．然而，如果我们可以改进实现算法G的特定计算机程序，使得它可以在1/2的时间内运行，或者如果我们可以在更快的计算机上实现G，使得它可以在1/2的时间内运行，那么$f(n)$和$g(n)$将是一样的．因为常量1/2与n无关，把这个常量系数的改进看成是算法实现的函数而非算法的函数并不是很牵强的事．在这一意义下，因为$f(n)/g(n)$等于一个常量，即$f(n)=kg(n)$，所以实际上，函数$f(n)$和$g(n)$可以认为是相同的．

现在，f是$O(g)$意味着(对于相对较大的n)$f(n)\leqslant kg(n)$．但是因为$kg(n)$和$g(n)$在实际应用中可以认为是相同的，所以$f(n)\leqslant kg(n)$表明在所有实际应用中$f(n)\leqslant g(n)$．因此，f是$O(g)$说明复杂度g的算法不比复杂度f的算法更有效．

在考证多项式边界的判断标准之前，下面的定理概括了一些基本结果：

⊖ 在初级课程中本节应该跳过．

⊜ 只想理解这一定义的读者可以跳过本小节的其余内容．

定理 2.12

(a)如果 c 是一个正常量，那么 f 是 $O(cf)$ 且 cf 是 $O(f)$.

(b)n 是 $O(n^2)$，n^2 是 $O(n^3)$，…，n^{p-1} 是 $O(n^p)$，…. 然而，n^p 不是 $O(n^{p-1})$.

(c)如果 $f(n)=a_qn^q+a_{q-1}n^{q-1}+\cdots+a_0$ 是一个 q 阶多项式，$a_q>0$ 且对于所有的 i，如果 $a_i\geqslant 0$，则 f 是 $O(n^q)$.

97

(d)如果 $c>1$ 且 $p\geqslant 0$，则 n^p 是 $O(c^n)$. 进一步地，c^n 不是 $O(n^p)$.

定理 2.11(a)表明，正如我们所假设的那样，可以认为复杂度 f 和 cf 的算法效率相同. (b)表明 p 的值越小 $O(n^p)$ 算法越有效. (c)表明多项式的阶告诉我们多项式算法的相对复杂度. (d)表明多项式算法总是比指数算法更有效. 这就是为什么多项式算法被认为是好的，而指数算法不被认为是好的原因. 定理 2.12 的结果在表 2.11 中得到了生动的说明，此表表明不同复杂度函数增长的快慢程度. 注意与其他复杂度函数比较，指数复杂度函数的变化要快的多.

表 2.11 不同复杂度函数的增长

输入大小 n	复杂度函数 $f(n)$			
	n	n^2	$10n^2$	2^n
5	5	25	250	32
10	10	10^2	10^3	$1024\approx 1.02\times 10^2$
20	20	400	4000	$1\,048\,576\approx 1.05\times 10^6$
30	30	900	9000	$\approx 1.07\times 10^9$
50	50	2500	25 000	$\approx 1.13\times 10^{15}$
$100=10^2$	10^2	10^4	10^5	$\approx 1.27\times 10^{30}$
$1000=10^3$	10^3	10^6	10^7	$>10^{300}$
$10\,000=10^4$	10^4	10^8	10^9	$>10^{3000}$

定理 2.12 的证明

(a)显然 cf 是 $O(f)$. 取 $k=c$. 接下来，f 是 $O(cf)$，因为对于所有的 n 有 $f(n)\leqslant(1/c)cf(n)$.

(b)因为对于 $n\geqslant 1$，有 $n^p\geqslant n^{p-1}$，n^{p-1} 是 $O(n^p)$. 现在，n^p 不是 $O(n^{p-1})$. 因为 $n^p\leqslant cn^{p-1}$ 只对 $n\leqslant c$ 成立.

(c)因为对于所有的 i 和 $n\geqslant 1$，有 $a_i\geqslant 0$ 和 $a_in^i\leqslant a_in^q$. 因此，对于 $n\geqslant 1$，有 $f(n)\leqslant(a_0+a_1+\cdots+a_q)n^q$.

(d)这是微积分和高等微积分的一般结果. 注意到当 $n\to\infty$ 时，$n^p/c^n\to 0$ 这一事实就可以得到该结果. 而通过运用(微积分中的)洛必达法则 p 次可得这一结果. 因为 $n\to\infty$ 时 $n^p/c^n\to 0$，对于足够大的 n，有 $n^p/c^n\to 0, n^p/c^n\leqslant k$. 类似的分析表明当 $n\to\infty$ 时，$c^n/n^p\to\infty$，所以对于 $n\geqslant r$ 和常量 k，$c^n\leqslant kn^p$ 不成立. ■

98

(d)的证明间接提到极限可以用于证明 f 是否是“g 的大 0”的事实. 注意到 $f(n)$ 和 $g(n)$ 不是 n 的一般函数，而是 n 的非负函数是很重要的. 这是因为它们度量了算法的复杂度或成本.

定理 2.13 如果 $g(n)>0$ 且 $\lim\limits_{n\to\infty}\dfrac{f(n)}{g(n)}$ 存在，则 f 是 $O(g)$.

证明 假设 $\lim\limits_{n\to\infty}\dfrac{f(n)}{g(n)}=L$. 那么对于任意的 $\varepsilon>0$，存在 $N>0$，使得对于所有的 $n>N$，$\left|\dfrac{f(n)}{g(n)}-L\right|<\varepsilon$ 成立. 注意

$$\left|\frac{f(n)}{g(n)}-L\right|<\varepsilon$$
$$\Downarrow$$
$$\frac{f(n)}{g(n)}-L<\varepsilon$$
$$\Downarrow$$
$$f(n)<\varepsilon g(n)+Lg(n)$$
$$\Downarrow$$
$$f(n)<(\varepsilon+L)g(n).$$

使用“大 O”的定义，通过令 $r=N$ 和 $k=\varepsilon+L$ 可以完成证明．注意，这一证明使用了 $g(n)>0$ 的事实．■

使用定理 2.13 和微积分中的洛必达法则，我们看到 $f(n)=7\log n$ 是 $O(n^5)$，因为

$$\lim_{n\to\infty}\frac{7\log n}{n^5}=\lim_{n\to\infty}\frac{7\left(\frac{1}{n}\right)}{5n^4}=\lim_{n\to\infty}\frac{7}{5n^5}=0.$$

如果定理 2.13 中的极限不存在，那么就得不到任何结论．考虑 $f(n)=\sin n+1$ 和 $g(n)=\cos n+3$．极限 $\lim\limits_{n\to\infty}\frac{\sin n+1}{\cos n+3}$ 不存在．然而易知，对于 $k=1$ 和所有 $n\geqslant 1$ 有 $\sin n+1\leqslant k(\cos n+3)$ 成立．因此，f 是 $O(g)$．另一方面，正如我们在定理 2.12(d)的证明中所看到的那样，当 $n\to\infty$ 时有 $\lim\limits_{n\to\infty}\frac{c^n}{n^p}\to\infty$，而这意味着对于 $n\geqslant r$ 和常量 k，$c^n\leqslant kn^p$ 不成立．

在结束本小节之前，我们应该再一次注意到：我们的结果依赖于复杂度 f 和 cf 算法之间的“等价”，这是关键的，我们的结果也依赖于输入的大小 n 相当大的思想．在实践中，复杂度为 $100n$ 的算法肯定不如复杂度为 n 的算法好．另外，对于小的 n 值，它也肯定不如复杂度为 2^n 的算法好．因此，在实践中，$O(n)$算法可能不如 $O(2^n)$算法好．必须仔细理解本节的结果及对多项式算法的强调．

99

2.18.2　NP 完全问题

在算法的研究中，区别确定性过程和非确定性过程是很方便的．可以把算法看成状态到状态的转移．*确定性算法*(deterministic algorithm)一次只能转移到一个新状态，而*非确定性算法*(nondeterministic)一次可以转移到若干个新状态．即非确定性算法能够同时揭示几种可能性．本书集中考虑确定性算法，且实际上，当我们使用算法一词时，我们意指确定性算法．存在复杂度是多项式的确定性算法的问题类称为 P．存在复杂度是多项式的非确定性算法的问题类称为 NP．显然 P 中的每一个问题也在 NP 中．至今为止，还没有人发现 NP 中的问题能被证明不在 P 中．然而，有许多 NP 中的已知问题也许在也许不在 P 中．其中的大部分问题非常普通而且看起来很难，对这些问题来说，寻找确定性算法非常重要．Cook[1971]发现了一个重要的事实：存在一些问题 L，通称 *NP 难题*(NP-hard)，它们具有下面的性质：如果 L 可以通过确定性多项式算法解决，那么 NP 中的每一个问题都能够通过确定性算法解决．例 2.10 所讨论的卖货郎问题就是这样的一个 NP 难题．实际上，这是一个 NP 完全问题(NP-complete problem)，一个属于类 NP 的 NP 难题．Karp[1972]指出存在很多 NP 完全问题．现在，很多人都怀疑存在非确定性多项式算法的每一个问题都有确定性多项式算法．因此，他们怀疑将来是否可以为诸如卖货郎问题这样的 NP 难题(NP-完全问题)找到确定性的多项式算法．因此，NP 难题(NP-完全问题)在现实意

义下是很困难的. 关于 NP 完全性的全面讨论可以参见 Garey 和 Johnson[1979]. 也可参见 Reingold, Nievergelt 和 Deo[1977].

因为现实世界的问题需要得到解决，所以当我们发现一个问题是 NP 完全或 NP 难题时，不能只停留在寻找一个解决方案. 我们采用折中的办法，例如，研究的问题可能是非 NP 难题的特殊情况解. 例如，我们可以考虑离开任意城市时，只有两个最便宜线路的卖货郎问题；或者，当离开一个城市时，只考虑 5 个最近的城市的卖货郎问题. 另外，我们也可以寻找一个接近于我们考虑的问题的解的好算法. 现今组合数学中一个日益重要的活动是寻找贴近一个问题的(最优)解决方案的好算法.

本节练习

100

1. 在下面每种情况下，确定 f 是否是 $O(g)$ 并根据定义证明你的答案.
   (a) $f=2^n$, $g=5^n$　(b) $f=6n+2/n$, $g=n^2$
   (c) $f=10n$, $g=n^2$　(d) $f=4^{2n}$, $g=4^{5n}-50$
   (e) $f=n^2+7$, $g=\frac{1}{5}n^4$　(f) $f=\cos n$, $g=4$
2. 使用极限确定 f 是否是 $O(g)$.
   (a) $f=3^n$, $g=4^n$　(b) $f=n+2/n$, $g=n^3$
   (c) $f=10n$, $g=n^2$　(d) $f=3^{2n}$, $g=3^5n-100$
   (e) $f=n!$, $g=e^n$[提示：回忆 2.3 节斯特林近似法]
3. 证明：如果 f 是 $O(h)$ 且 g 是 $O(h)$，那么 $f+g$ 是 $O(h)$.
4. 证明：如果 f 是 $O(g)$ 且 g 是 $O(h)$，那么 f 是 $O(h)$.
5. 证明：如果 f 是 $O(g)$ 且 g 是 $O(f)$，那么对于所有的 h，h 是 $O(f)$ 当且仅当 h 是 $O(g)$.
6. (a)证明 $\log_2 n$ 是 $O(n)$.　(b)证明 n 不是 $O(\log_2 n)$.
7. 假设 $c>1$.
   (a)证明 c^n 是 $O(n!)$.　(b)证明 $n!$ 不是 $O(c^n)$.
8. (a) n 是 $O(n\log_2 n)$ 成立吗？为什么？
   (b) $n\log_2 n$ 是 $O(n)$ 成立吗？为什么？
9. 对于下面的每一个函数 f 和 g，确定 f 是否是 $O(g)$ 并证明你的答案. 你可以使用本节中的定义、极限和定理及前面练习中的结果.
   (a) $f=5n$, $g=n^3$　(b) $f=n^3$, $g=5n$
   (c) $f=6n^2+2n+1$, $g=n^2$　(d) $f=n^3$, $g=2^n$
   (e) $f=n^3$, $g=n^5$　(f) $f=7n^7$, $g=3^n$
   (g) $f=10\log_2 n$, $g=n$　(h) $f=n^2+2^n$, $g=4^n$
   (i) $f=\log_2 n+25$, $g=n^3$　(j) $f=n^2$, $g=2^n+n$
   (k) $f=7^n$, $g=4n^{50}+25n^{10}+100n$　(l) $f=3n^4+2n^2+1$, $g=2^n$
   (m) $f=n+n\log_2 n$, $g=2^n$　(n) $f=n^3+3^n$, $g=n!$
10. 解释什么函数是 $O(1)$，即对于 $g\equiv 1$，什么函数是 $O(g)$？
11. 令 $f(n)=\sum_{i=1}^{n} i$. 证明 f 是 $O(n^2)$.
12. 下面的函数中哪些是其他函数的“大 O”：$4n\log_2 n+\log_2 n$、n^2+3/n^2、$\log_2\log_2 n$.
13. 对下面的函数重复练习 12：$(\log_2 n)^2$，2^n 和 $4n^2\log_2 n+5n$.

2.19 鸽巢原理及其扩展

2.19.1 最简单的鸽巢原理

在组合数学中，证明某种排列或模式存在的一个应用最广泛的工具是*鸽巢原理*(pigeonhole principle). 通俗地，这一原理叙述如下：如果存在"很多"鸽子和"少数"鸽巢，那么必定有两个或更多只鸽子占据同一个鸽巢. 这一原理也称*狄利克雷抽屉原理*(Dirichlet drawer principle)[⊖]或*鞋盒原理*(shoebox principle)，它还有一些其他名字. 即如果存在很多对象(鞋)和少数抽屉(鞋盒)，那么某个抽屉(鞋盒)必须有两个或两个以上的对象(鞋). 我们给出这一基本组合数学原理的几种变形和这一原理的若干应用. 注意，鸽巢原理只是说一定存在两只或两只以上的鸽子占据相同的鸽巢. 这对我们识别这些鸽子没有帮助.

101

首先我们更精确地描述鸽巢原理.

定理 2.14(鸽巢原理) *如果把 $k+1$ 只鸽子放入 k 个鸽巢里，那么至少有一个鸽巢包含两只或更多只鸽子.*

为了说明定理 2.14，我们注意到如果在一个房间里有 13 个人，那么他们中至少有两个人的生日在同一月份. 类似地，如果从电话薄上选出 677 个人，那么他们中至少有两个人的名和姓由同样的字母开始. 下面是两个更深入的例子.

【例 2.33 调度立法委员会的会议(再探例 1.4)】 考虑例 1.4 的会议调度问题. 一个团(clique)是由委员会集合组成的，其中的每一对都有一名公共的成员. 对应于这一委员会集合的团数(clique number)是最大团的大小. 已知表 1.5 的数据，最大团的大小是 3. 大小是 3 的团对应于图 1.1 中的三角形. 因为在一个团中的所有委员会必须得到不同的会议时间，所以鸽巢原理说明所需的会议时间的数量至少与最大团的大小一样大. 为了明白其原因，令一个团的顶点是鸽子，而会议时间是鸽巢(用第 3 章的语言，这一结论说明图的颜色数至少总是与团数一样多). ■

【例 2.34 制作个人计算机】 个人计算机(PC)的制造商在 30 天的期间内每天至少制作一台 PC，并在完成一台 PC 后才能开始下一台 PC 的制作工作，且每天制作的 PC 平均不超过 $1\frac{1}{2}$ 台. 那么一定存在连续的若干天，在这些天正好制作了 14 台 PC. 为了明白其原因，设 a_i 是第 i 天结束时 PC 的累计制作数量. 因为至少每天生产一台 PC，而且 30 天内至多生产 45 台 PC. 我们有

$$a_1 < a_2 < \cdots < a_{30},$$
$$a_1 \geqslant 1,$$
$$a_{30} \leqslant 45.$$

还有

102

$$a_1 + 14 < a_2 + 14 < \cdots < a_{30} + 14 \leqslant 45 + 14 = 59.$$

现在考虑下面这些数：

$$a_1, a_2, \cdots, a_{30}, a_1 + 14, a_2 + 14, \cdots, a_{30} + 14,$$

共有 60 个数，每一个数都在 1 和 59 之间. 根据鸽巢原理，其中有两个数是相等的. 因为 a_1, a_2, …, a_{30}都不相同且 a_1+14, a_2+14, …, $a_{30}+14$ 也都不相同，所以存在 i 和 j $(i\neq j)$使得 $a_i=a_j+14$. 因此，在日期 i 和 j 之间，制造商正好制作了 14 台 PC. ■

⊖ 尽管鸽巢原理的来源不太清楚，但是 19 世纪的数学家彼得 · 狄利克雷广泛地使用了它.

2.19.2　鸽巢原理的扩展和应用

我们从给出鸽巢原理的若干更强版本开始本小节．特别地，当把 $2k+1$ 只鸽子放入 k 个鸽巢时，至少有一个鸽巢里的鸽子数超过二．这一结果成立，因为如果每一个鸽巢里至多有两只鸽子，那么因为有 k 个鸽巢，所以总共至多有 $2k$ 只鸽子．根据相同的推理，把 $3k+1$ 只鸽子放入 k 个鸽巢时，至少有一个鸽巢里的鸽子数超过三．

一般地，我们有下面的定理．

定理 2.15　当把 m 只鸽子放入 k 个鸽巢时，至少有一个鸽巢里的鸽子数超过

$$\left\lfloor\frac{m-1}{k}\right\rfloor.$$

证明　如果鸽巢中的鸽子数至多是 $\lfloor(m-1)/k\rfloor$，那么鸽子的总数至多是

$$k\left\lfloor\frac{m-1}{k}\right\rfloor\leqslant m-1<m.$$

■

为了说明定理 2.15，注意，如果在一个房间里有 40 个人，那么至少有 3 个人的生日在同一月份，因为 $\lfloor 39/12\rfloor=3$．作为上述结果的另一个应用，假设我们知道一台计算机内有八个存储位置，共有 8000 位存储容量．那么我们知道至少一个存储位置的存储容量不低于 1000 位．因为 $m=8000$ 且 $k=8$，所以有 $\lfloor(m-1)/k\rfloor=999$．类似地，如果一个工厂有 40 个电插座共拥有 9600 伏特的容量，那么我们知道至少存在一个拥有 240 伏特或更高伏特容量的电插座．

后面的两个例子说明定理 2.15 的如下推论，其形式证明留给读者(练习 21)．

推论 2.15.1　一个数集的平均值在这些数的最小数和最大数之间．

103

这一推论的另一种形式如下．

推论 2.15.2　给定一个数集，在这个集合中总是存在一个数，它的值至少与这个集合中的数的平均值一样大(一样小)．

【例 2.35 调度立法委员会的会议时间(再探例 2.33)】　继续例 2.33．委员会的集合是一个独立集合(independent set)，如果其中的每一对委员会都没有共同成员．一个会议时间安排把 n 个委员会分割成 k 个不相交的小组，其中每个小组都是独立集合．这样的独立集合的平均大小为 n/k．因此，至少存在一个大小至少是 n/k 的独立集合．在表 1.5 的数据中，$n=6$，$k=3$ 且 $n/k=2$．如果有 8 个委员会和 3 个会议时间，我们知道至少有一个会议时间将至少有 8/3 个委员会，即至少有 3 个委员会． ■

【例 2.36 网络服务器】　一家公司有 15 台网络服务器和 10 个因特网端口．任意时刻都不会有超过 10 台的服务器同时需要端口．每 5 分钟，这些服务器的某个子集请求端口．我们希望以这样的方式把每一个服务器与某些端口相连接：使用尽可能少的连接，总是确保一台网络服务器有一个存取端口(一个端口一次至多被一个服务器使用)．这时，需要多少次连接呢？为了回答这一问题，注意，如果连接少于 60 次，那么平均一个端口的连接少于 6 次，所以根据推论 2.15.2，某个端口将与 5 台或更少的服务器连接．如果余下的 10 台网络服务器同时使用，那么只有 9 个端口供它们使用．因此，至少需要 60 次连接．证明存在有所需性质的 60 次连接的排列留给读者(练习 19)． ■

定理 2.15 的另一个应用是关于数序列的递增和递减子序列的结果．考虑数序列 x_1，x_2，$\cdots$，x_p．一个子序列(subsequence)是使得 $1\leqslant i_1<i_2<\cdots<i_p\leqslant p$ 成立的任意序列 x_{i_1}，$x_{i_2}\cdots x_{i_q}$．例如，$x_1=9$，$x_2=6$，$x_3=14$，$x_4=8$，$x_5=17$，那么我们有序列 9，6，14，8，17；子序列 x_2，x_4，x_5 是序列 6，8，17；子序列 x_1，x_3，x_4，x_5 是序列 9，14，8，17 等．一个子序列

是递增的(increasing)，如果它各项的值相继增加；子序列是递减的(decreasing)，如果它各项的值相继减少. 在我们的例子中，最长的递增子序列是 9，14，17，最长的递减子序列是 14，8.

作为另一个例子，考虑序列：

$$12, 5, 4, 3, 8, 7, 6, 11, 10, 9.$$

最长的递增子序列是 5，8，11，而最长的递减子序列是 12，11，10，9. 这两个例子说明下面的定理，它的证明依赖于定理 2. 15.

104

定理 2.16(Erdös 和 Szekeres[1935])　给定一个有 n^2+1 个不同整数的序列，或者存在有 $n+1$ 个项的递增子序列或者存在有 $n+1$ 个项的递减子序列.

注意这一定理需要有 n^2+1 个项；即对于项数小于 n^2+1 的整数序列结论可能不成立. 例如，考虑序列

$$3, 2, 1, 6, 5, 4, 9, 8, 7.$$

这是一个项数为 9 的整数序列，其中最长的递增子序列和最长的递减子序列有 3 个项.

定理 2.16 的证明　设这一序列是

$$x_1, x_2, \cdots, x_{n^2+1}.$$

令 t_i 是开始于 x_i 的最长的递增子序列的项数. 如果任意 t_i 至少是 $n+1$，那么定理得到证明. 因此，假设每一个 t_i 在 1 与 n 之间. 因此，我们有 n^2+1(n^2+1 个 t_i)只鸽子要被放入 n 个鸽巢(数 1，2，…，n)中. 根据定理 2.15，存在一个至少有

$$\left\lfloor\frac{(n^2+1)-1}{n}\right\rfloor+1=n+1$$

只鸽子的鸽巢. 即至少有 $n+1$ 个相等的 t_i. 证明与这些 t_i 相关联的 x_i 形成一个递减子序列. 我们证明：假设对 $i<j$ 有 $t_i=t_j$，则 $x_i>x_j$. 如果 $x_i\leqslant x_j$，那么有 $x_i<x_j$，因为根据假设 n^2+1 个整数都不相同. 这时，x_i 及其后开始于 x_j 的最长递增子序列形成一个长度为 t_j+1 的递增子序列. 因此有 $t_i\geqslant t_j+1$. 这是一个矛盾. ■

为了说明这一证明，考虑下面 10 个不同整数的序列：

$$10, 3, 2, 1, 6, 5, 4, 9, 8, 7.$$

这里 $n=3$ 因为 $10=3^2+1$，且我们有

i	x_i	t_i	样本子序列
1	10	1	10
2	3	3	3，6，7
3	2	3	2，6，7
4	1	3	1，6，7
5	6	2	6，7

i	x_i	t_i	样本子序列
6	5	2	5，7
7	4	2	4，7
8	9	1	9
9	8	1	8
10	7	1	7

因此，有四个等于 1 的 t_i，相对应的 x_i，即 x_1，x_8，x_9，x_{10}形成一个递减子序列 10，9，8，7.

105

我们给出鸽巢原理的一个更一般形式来结束本节，它的证明留给读者(练习 22).

定理 2.17　假设 p_1，p_2，…，p_k 是正整数. 如果把

$$p_1+p_2+\cdots+p_k-k+1$$

只鸽子放入 k 个鸽巢中，那么或者第一个鸽巢至少有 p_1 只鸽子，或者第二个鸽巢至少有 p_2 只鸽子，……，或者第 k 个鸽巢至少有 p_k 只鸽子.

2.19.3 拉姆齐数

定理 2.15 中所陈述的鸽巢原理形式的一个简单应用就是如下定理.

定理 2.18 *假设在 6 个人中每两个人或者是朋友或者是敌人. 那么或者有 3 个人互为朋友或者有 3 个人互为敌人.*

证明 设 a 是其中的任意一个人. 根据鸽巢原理, 剩余的 5 个人中, 或者 3 个或更多人是 a 的朋友, 或者 3 个或更多人是 a 的敌人(在定理 2.15 中取 $m=5$, $k=2$). 首先假设 b, c 和 d 是 a 的朋友. 如果这些人中任意 2 个人是朋友, 那么这 2 个人和 a 形成一个互为朋友的 3 人组. 如果 b, c 和 d 都不是朋友, 那么 b, c 和 d 形成一个互为敌人的 3 人组. 假设 b, c 和 d 是 a 的敌人时的论证也类似. ■

定理 2.18 是组合数学课题中称为拉姆齐理论的最简单结果, 这一理论要追溯到拉姆齐的原始文章[1930]. 它可以重述如下:

定理 2.19 *假设 S 是 6 个元素的任意集合. 如果我们把 S 的 2 子集分成 X 和 Y 两类, 那么或者*

1. *存在 S 的 3 子集, 它的所有 2 子集都在 X 中, 或者*
2. *存在 S 的 3 子集, 它的所有 2 子集都在 Y 中.*

扩展这一结论, 假设 p 和 q 是满足 $p, q\geqslant 2$ 的整数. 我们说正整数 N 有 (p, q) *拉姆齐性质*(Ramsey property)如果下面的条件成立: 给定 N 个元素的任意集合 S, 如果我们把 S 的 2 子集分成 X 和 Y 两类, 那么或者

1. 存在 S 的 p 子集, 其所有 2 子集都在 X 内, 或者
2. 存在 S 的 q 子集, 其所有 2 子集都在 Y 内.

例如, 根据定理 2.19, 数 6 有(3, 3)拉姆齐性质. 然而数值 3 就不具有(3, 3)拉姆齐性质. 因为考虑集合 $S=\{a, b, c\}$ 并把 S 的 2 子集分类成 $X=\{\{a, b\}, \{b, c\}\}$ 和 $Y=\{\{a, c\}\}$. 那么显然, 不存在 S 的 3 子集, 其所有 2 子集都在 X 中或存在 S 的 3 子集, 其所有 2 子集都在 Y 内. 类似地, 数 4, 5 也不具有(3, 3)拉姆齐性质. [106]

注意 *如果数 N 有 (q, p) 拉姆齐性质且 $M>N$, 那么数 M 有 (p, q) 拉姆齐性质(为什么). 拉姆齐理论的主要定理说明拉姆齐性质的定义是恰当的.*

定理 2.20(拉姆齐定理[⊖]) *如果 p 和 q 是满足 $p, q\geqslant 2$ 的整数, 那么存在有 (p, q) 拉姆齐性质的正整数 N.*

对于定理的证明, 可以参见 Graham, Rothschild 和 Spencer[1990].

拉姆齐理论的一个关键问题是寻找*拉姆齐数*(Ramsey number)$R(p, q)$, 这是具有 (p, q) 拉姆齐性质的最小的数. 注意, 根据定理 2.19, $R(3, 3)\leqslant 6$. 事实上, $R(3, 3)=6$. 计算拉姆齐数 $R(p, q)$ 的问题是优化问题的例子. 所以, 在计算拉姆齐数的尝试中, 我们是在处理组合数学的第三类基本问题, 即优化问题. 拉姆齐数计算一般来说是一个难题. 只有少数几个拉姆齐数是明确已知的. 实际上, 只有表 2.12 给出的那些已知的拉姆齐数 $R(p, q)$(3.8 节给出其中某些项的证明或部分证明). 关于拉姆齐数的更全面的概括性文章, 向读者推荐 Radziszowski[2002]. 也可参见 Graham[1981]、Graham, Rothschild 和 Spencer[1990]、Chung 和 Grinstead[1983].

⊖ 这一定理实质上包含在拉姆齐的原始文章[1930]中, 这篇文章主要涉及的是在形式逻辑中的应用. 其基本结果由 Erdös 和 Szekeres[1935]重新发现并推广. 关于其说明可以参见 Graham, Rothschild 和 Spencer[1990].

表 2.12　已知的拉姆齐数 $R(p, q)$①

p	q	$R(p, q)$	参考文献(s)
2	n	n	
3	3	6	Greenwood 和 Gleason[1955]
3	4	9	Greenwood 和 Gleason[1955]
3	5	14	Greenwood 和 Gleason[1955]
3	6	18	Kéry[1964]
3	7	23	Kalbfleisch[1966]，Graver 和 Yackel[1968]
3	8	28	Grinstead 和 Roberts[1982]、McKay 和 Min[1992]
3	9	36	Kalbfleisch[1966]、Grinstead 和 Roberts[1982]
4	4	18	Greenwood 和 Gleason[1955]
4	5	25	Kalbfleisch[1965]、McKay 和 Radziszowski[1995]

注：①注意有 $R(p, q)=R(q, p)$.

拉姆齐定理(定理 2.20)有各种扩展形式. 有一些将在 3.8 节讨论. 对于其他扩展形式，参见 Graham，Rothschild 和 Spencer[1990].

拉姆齐理论在诸如噪声信道干扰图、包交换网络的设计、信息检索及决策等课题中有一些令人感兴趣的应用. 我们将在 4.3.3 节对决策应用加以讨论. 关于拉姆齐理论的其他应用的概况可以参见 Roberts[1984].

107

本节练习

1. 必须选出多少人才能确保至少两个人有下面的性质：
   (a)名字中的第一个字母相同？
   (b)生在一年的同一天？
   (c)身份证的后四个数字相同？
   (d)电话号码的前三个数字相同？
2. 如果要求至少有三个人有所要求的性质，重复练习 1.
3. 如果有五双不同的袜子混放在一个抽屉里，在我们能够肯定找到一双袜子之前，必须选出多少只袜子？
4. (a)我们必须选出多少个三数字位串才能确保其中的两个至少有一位数字相同？
   (b)我们必须选出多少个 n 数字位串才能确保其中的两个至少有一位数字相同？
5. 要对 301 门课程安排期末考试时间，使得有共同学生的两门课程得到不同的考试时间，且有 20 个考试时间就足够了. 你能说明如何确定在任意一个时间可安排的课程的最大数量吗？
6. 如果一家汽车租赁公司有 95 辆汽车，总共有 465 个座位. 你能肯定存在至少有 5 个座位的汽车吗？
7. 如果一所学校有 400 门课程，每门课程平均有 40 个学生，对学生最多的一门课程你能得出什么结论？
8. 如果一个电话切换网络有 20 个切换站，平均每个切换站有 65 000 个连接. 你能说出最小的切换站的连接数量是多少吗？
9. 一个比萨派对上留有 3 张橄榄比萨、5 张普通比萨、7 张意大利辣肠比萨和 8 张凤尾鱼比萨.
   (a)要求有多少张比萨才能确保至少一种比萨有三张？
   (b)要求有多少张比萨才能确保得到 5 张凤尾鱼比萨？
10. 一名建筑监理有 77 天做巡回检查. 他想一天至少做一个检查，且要做 132 个检查. 存在他刚好做 21 个检查的连续日周期吗？为什么？
11. 一名研究员要在 50 天的周期内每天至少做一个试验，而总试验次数不超过 75.
    (a)证明在这 50 天内，存在正好做 24 个试验的连续日子.
    (b)如果要做 30 个试验，这一结论仍然正确吗？
12. 给出一个委员会调度问题的例子，其中最大派系的大小比所需的会议时间数小.

108

13. 寻找下面每一个序列的最长递增子序列和最长递减子序列并检验你的结论符合 Erdös-Szekeres 定理.

(a)6，5，7，4，1　　　(b)6，5，7，4，1，10，9，11，14，3

(c)4，12，3，7，14，13，15，16，10，8

14. 给出这样的序列：含 16 个不同整数且即没有含 5 个项的递增子序列也没有含 5 个项的递减子序列.
15. 一名雇员的出勤钟表明她在 10 天的周期内工作了 81 个小时. 证明在某个连续的两天内，她工作了至少 17 个小时.
16. 在 15 天的周期内，调制解调器连接使用了 300 个小时. 证明在连续 3 天的某个周期内，这个调制解调器至少使用了 60 个小时.
17. 一个公司有 25 名工人共用 12 台切割机. 每一小时，某组工人需要一台切割机. 在任意给定时间需要切割机的工人从不超过 12 名. 我们给每一台机器做一张准许使用该机器的工人列表，而且保证每一名工人至少在一张机器列表上. 如果把 12 台机器的列表上的人数累加起来，总数是 95. 证明：在某个时间某名工人没有可用的机器.
18. 考虑下面的序列：

$$9,4,7,3,2,8,10,5,6,1.$$

寻找定理 2.16 的证明中所定义的数 t_i，并使用这些 t_i 寻找至少含 4 个项的递减子序列.
19. 在例 2.36 中，证明存在具有期望性质且有 60 个连接的安排.
20. 假设在一个派对上有 10 个人，他们的年龄(整数)范围从 0 到 100.

(a)证明存在两个不同的，但不必不相交的的子集，这两个子集中人的年龄总数正好相同.

(b)使用(a)的两个子集，证明存在两个不相交的子集，其中的人的年龄总数恰好相同.
21. 由定理 2.15 证明推论 2.15.1.
22. 证明定理 2.17.
23. 证明：如果从集合 1，2，3，…，$2n$ 中选出 $n+1$ 个数，那么其中一个数将整除另外一个数.
24. 证明：在至少有 2 个人的一个小组中，总是存在 2 个人，他们在这个小组中有相同数量的熟人.
25. 一位面试官要给每一名工作申请人一个 P(通过)或 F(未通过)的面试评估. 她发现无论她如何安排这些评估，总是至少有 3 个人得到相同的评估. 她需要面试的申请人的最小数目是多少？
26. 如果总是至少有 4 个人有相同的评估，重复练习 25.
27. 给定 p 个整数 a_1，a_2，…，a_p 的序列，证明在这个序列中存在连续几项，它们的和可以被 p 整除. 即证明存在 i 和 j，$1\leqslant i\leqslant j\leqslant p$，使得 $a_i+a_{i+1}+\cdots+a_j$ 可以被 p 整除.　[109]
28. 证明：给定 $R(n+1，n+1)$ 个不同整数的序列，或者存在 $n+1$ 个项的递增子序列或者存在 $n+1$ 个项的递减子序列.
29. 通过给出 X 和 Y 的划分，证明：

(a)数 4 没有(3，3)拉姆齐性质.

(b)数 5 没有(3，4)拉姆齐性质.

(c)数 6 没有(4，4)拉姆齐性质.
30. 求下面的拉姆齐数：

(a)$R(2，2)$　　(b)$R(2，8)$　　(c)$R(7，2)$
31. 证明：如果数 N 有$(p，q)$拉姆齐性质，且 $M>N$，那么数 M 有$(p，q)$拉姆齐性质.
32. 考虑 10 人的一个小组，其中每两个人或者是朋友或者是敌人.

(a)证明：如果这一小组中的某个人至少有 4 个朋友，那么或者有 3 个互为朋友的人，或者有 4 个互为敌人的人.

(b)类似地，如果在这个小组中某个人至少有 6 个敌人，证明或者有 3 个互为朋友的人，或者有 4 个互为敌人的人.

(c)根据(a)和(b)证明，在 10 个人的一个小组中，如果每两个人或者是朋友或者是敌人，那么或者有 3 个互为朋友的人，或者有 4 个互为敌人的人.

(d)关于拉姆齐数(c)说明了什么？

33. 假设 p，q 和 r 是整数，且 $p\geqslant r$，$q\geqslant r$，$r\geqslant 1$. 正整数 N 有 (p, q, r) 拉姆齐性质，如果下列条件成立：给定 N 个元素的任意集合 S，如果我们把 S 的 r 子集分成 X 和 Y 两类，那么或者：

    1. 存在 S 的 p 子集，它的所有 r 子集都在 X 内，或
    2. 存在 S 的 q 子集，它的所有 r 子集都在 Y 内.

    拉姆齐数 $R(p, q; r)$ 可定义为有 $(p, q; r)$ 拉姆齐性质的最小整数 N.[⊖] 对于 $R(p, q; r)$ 有良好定义的证明，即总存在这样的整数 N 的证明，可以参见 Graham，Rothschild 和 Spencer[1990].

    (a)证明 $R(p, q; 1)=p+q-1$

    110 (b)证明 $R(p, r; r)=p$ 且 $R(r, q; r)=q$.

附加练习

1. 有 1000 名申请人申请入学资格，而该学院计划接纳 300 人. 对于这个学院来说存在多少种选出 300 人的可能方法？
2. 八肽(octapeptide)是由 8 个氨基酸组成的链，其中每一个氨基酸是 20 种自然存在的氨基酸中的一个. 存在多少种八肽？
3. 在 15 个基的 RNA 链中，有 3 个 A、6 个 U、5 个 G 和 1 个 C. 如果这个链开始于 GU 且结束于 ACU，那么有多少种这样的链？
4. 有多少个把数 0 或 1 指定给每一个由 0 和 1 组成的 $m\times n$ 矩阵的函数？
5. 有多少个这样的 5 变量开关函数，它或者把 1 指定给开始于 1 的所有位串，或把 0 指定给开始于 1 的所有位串？
6. 在安排用具修理时，有八家被指定在上午，九家被指定在下午. 我们能够安排的修理顺序有多少种？
7. 如果校园电话内线有 4 个数字且没有重复数字，有多少种可能的不同内线电话？
8. 在 20 个基的 RNA 链中，有 4 个 A、5 个 U、6 个 G 和 5 个 C. 如果这个链或开始于 AC 或 UG，那么这样的链有多少种？
9. 单词 reconnaissance 的字母有多少种可区分排列？
10. 20 个氨基酸的链中有 5 个组氨酸、6 个精氨酸、4 个糖胶、1 个冬酰胺酶、3 个赖氨酸和 1 个谷氨酸. 这样的链有多少种？
11. 有一个由 10 个组成部分组成的系统. 当前 5 个组成部分中至少有 4 个运转且后 5 个组成部分中至少有 4 个运转时，这一系统运转. 这一系统可以运转的方式有多少种？
12. 15 种刷油漆的工作必须在一天内完成，其中 5 个工作时间短，4 个工作时间长，6 个工作时间为中等. 如果这 15 种工作都是可以区分的，有多少种不同的运作顺序，使得：

    (a)所有时间短的工作都在开始时进行？

    (b)所有花费时间相同的工作都连续进行？
13. 一个人希望访问 6 个城市，每一个城市正好访问两次，且从不连续访问同一城市两次. 有多少种实现这一方案的方法？
14. 有 9 个孩子的家庭中，有 2 个孩子有黑头发、3 个孩子有棕色头发、1 个孩子有红头发、3 个孩子有金发. 这样的家庭有多少种出生顺序？
15. 一家冷饮店提供 29 种不同的风味. 现要制作 3 球圆筒冰淇淋. 如果圆筒中的每一个球必须是风味不同的冰淇淋，那么有多少种不同的 3 球圆筒冰淇淋？
16. 一个人有 6 套不同的套装，他能用多少种方法选出不是一套的夹克和裤子？
17. 假设在街区有 11 座房子，其中 6 座有白蚁.

    111 (a)有白蚁的房子都彼此相邻的方式有多少种？

    (b)有白蚁的房子彼此都不相邻的方式有多少种？

⊖ (p, q) 拉姆齐性质与 $(p, q; 2)$ 拉姆齐性质相同.

18. 把 8 件礼品分给 5 个孩子，有多少种分配方法？
19. 有这样的一个实验：给 5 个实验者每个人一些药丸吃，如果有 20 个药丸且每一个实验者必须至少吃一个药丸. 有多少种方法进行这样的实验？
20. 如果有 75 种食谱可供选择，如果不记选择食谱的顺序，且可以有相同的食谱，那么有多少种方法在 7 天的期间准备 20 种食谱？
21. 一个人有 5 种共有基金，总共持有 56 个股份. 关于最大的基金中的股份数说明了什么？
22. 一个网站在 15 天期间被"点击"300 次. 证明在某个 3 天的连续期间，它至少被点击 60 次.
23. 在下面的条件下，把排列 84316275 变成恒等排列需要多少步？
    (a)只允许交换第 i 个元素和第$(i+1)$个元素？
    (b)只允许用一个子序列(如 3162)的倒位序列(2613)取代这一子序列？
24. (Pólya)考虑有图 2.7 所示的街道栅格的城市. 在一个实验中，每一个接受实验的人在拐角 A 出发，且被告知前进到拐角 B，B 位于从 A 向东五个街区、向北四个街区. 给他指定的路线正好走九个街区. 这个实验人想用 100 名接受实验的人，并且为每一个人指定不同的路线. 有可能完成这一实验吗？为什么？

B

A

图 2.7 城市街道栅格. 虚线给出一个典型路线

25. (Liu[1968])我们在纸条上印一个 5 位数，包含开始于 0 的数，如 00158. 因为数字 0，1 和 8 上下看都一样，而当被颠倒时，数字 6 和 9 被交换，所以如 61891 和 16819 这样的 5 位数可以分享同一纸条. 如果我们想包含所有可能的 5 位数，那么我们需要多少片不同的纸条？

112

参考文献

APPLEGATE, D., BIXBY, R., CHVÁTAL, V., and COOK, W., "On the Solution of Traveling Salesman Problems," *Doc. Math., Extra Vol. III*, Proceedings of the ICM (1998), 645–656.

AUMANN, R. J., and HART, S. (eds.), *Handbook of Game Theory with Economic Applications*, Elsevier Science, New York, 1998.

BAFNA, V., and PEVZNER, P. A., "Genome Rearrangements and Sorting by Reversal," *SIAM J. Comput., 25* (1996), 272–289.

BAFNA, V., and PEVZNER, P. A., "Sorting by Transpositions," *SIAM J. Discrete Math., 11* (1998), 224–240.

BALL, M. O., COLBOURN, C. J., and PROVAN, J. S., "Network Reliability," in M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser (eds.), *Handbooks in Operations Research and Management Science, 7*, North-Holland Publishing Co., Amsterdam, 1995, 673–762.

BANZHAF, J. F., III, "Weighted Voting Doesn't Work: A Mathematical Analysis," *Rutgers Law Rev., 19* (1965), 317–343.

BARLOW, R. E., and PROSCHAN, F., *Statistical Theory of Reliability and Life Testing: Probability Models*, Holt, Rinehart and Winston, New York, 1975.

BERGE, C., *Principles of Combinatorics*, Academic Press, New York, 1971.

BLAND, R. G., and SHALLCROSS, D. F., "Large Traveling Salesman Problems Arising from Experiments in X-Ray Crystallography: A Preliminary Report on Computation," *Oper. Res. Letters, 8* (1989), 125–128.

BOROS, E., GURVICH, V., HAMMER, P. L., IBARAKI, T., and KOGAN, A., "Decomposability of Partially Defined Boolean Functions," *Discrete Appl. Math., 62* (1995), 51–75.

BOROS, E., IBARAKI, T., and MAKINO, K., "Error-free and Best-Fit Extensions of Partially Defined Boolean Functions," *Inform. Comput., 140* (1998), 254–283.

BOROS, E., HAMMER, P. L., IBARAKI, T., KOGAN, A., MAYORAZ, E., and MUCHNIK, I., "An Implementation of Logical Analysis of Data," *IEEE Trans. Knowl. Data Eng.*, *12* (2000), 292–306.

BOYCE, D. M., and CROSS, M. J., "An Algorithm for the Shapley-Shubik Voting Power Index for Weighted Voting," unpublished Bell Telephone Laboratories manuscript, 1973.

BRAMS, S. J., *Voting Procedures*, North-Holland, Amsterdam, 1994.

BRAMS, S. J., and FISHBURN, P. C., *Approval Voting*, Birkhauser, Boston, 1983.

BRAMS, S. J., LUCAS, W. F., and STRAFFIN, P. D. (eds.), *Political and Related Models*, Vol. 2 of *Modules in Applied Mathematics*, Springer-Verlag, New York, 1983.

BRAMS, S. J., SCHOTTER, A., and SCHWÖDIANER, G. (eds.), *Applied Game Theory*, IHS-Studies No. 1, Physica-Verlag, Würzburg, 1979.

BUCK, R. C., *Advanced Calculus*, 2nd ed., McGraw-Hill, New York, 1965.

CAPRARA, A., "Sorting by Reversals Is Difficult," *Proc. of RECOMB 97: The First International Conf. on Comp. Biology*, ACM Press, New York, 1997, 75–83.

CHRISTIE, D. A., "Sorting Permutations by Block-Interchanges," *Inform. Process. Lett.*, *60* (1996), 165–169.

CHUNG, F. R. K., and GRINSTEAD, C. M., "A Survey of Bounds for Classical Ramsey Numbers," *J. Graph Theory*, *7* (1983), 25–37.

CLOTE, P., and BACKOFEN, R., *Computational Molecular Biology: An Introduction*, Wiley, New York, 2000.

113

COHEN, D. I. A., *Basic Techniques of Combinatorial Theory*, Wiley, New York, 1978.

COLEMAN, J. S., "Control of Collectivities and the Power of a Collectivity to Act," in B. Lieberman (ed.), *Social Choice*, Gordon and Breach, New York, 1971, 269–298.

CONGRESS OF THE UNITED STATES, OFFICE OF TECHNOLOGY ASSESSMENT STAFF, *Mapping Our Genes; Genome Projects - How Big, How Fast?*, Johns Hopkins University Press, Baltimore, MD, 1988.

COOK, S. A., "The Complexity of Theorem-Proving Procedures," *Proceedings of the Third ACM Symposium on Theory of Computing*, Association for Computing Machinery, New York, 1971, 151–158.

CRAMA, Y., HAMMER, P. L., and IBARAKI, T., "Cause-Effect Relationships and Partially Defined Boolean Functions," *Annals of Oper. Res.*, *16* (1988), 299–325.

CUMMINGS, M. R., *Human Heredity: Principles and Issues*, West Publishing Company, St. Paul, MN, 1997.

DEEGAN, J., and PACKEL, E. W., "A New Index of Power for Simple n-Person Games," *International Journal of Game Theory*, *7* (1978), 113–123.

DEMBSKI, W. A., *The Design Inference: Eliminating Chance through Small Probabilities*, Cambridge University Press, New York, 1998.

DEO, N., *Graph Theory with Applications to Engineering and Computer Science*, Prentice Hall, Englewood Cliffs, NJ, 1974.

DOBZHANSKY, T., *Genetics of the Evolutionary Process*, Columbia University Press, New York, 1970.

EDMONDS, J., "Paths, Trees, and Flowers," *Canad. J. Math.*, *17* (1965), 449–467.

EKIN, O., HAMMER, P. L., and KOGAN, A., "Convexity and Logical Analysis of Data," *Theoret. Comput. Sci.*, *244* (2000), 95–116.

ELSAYED, E. A., "Algorithms for Optimal Material Handling in Automatic Warehousing Systems," *Int. J. Prod. Res.*, *19* (1981), 525–535.

ELSAYED, E. A., and STERN, R. G., "Computerized Algorithms for Order Processing in Automated Warehousing Systems," *Int. J. Prod. Res.*, *21* (1983), 579–586.

ERDŐS, P., and SZEKERES, G., "A Combinatorial Problem in Geometry," *Composito Math.*, *2* (1935), 464–470.

FARACH-COLTON, M., ROBERTS, F. S., VINGRON, M., and WATERMAN, M. S. (eds.), *Mathematical Support for Molecular Biology*, Vol. 47, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, Providence, RI, 1999.

FARQUHARSON, R., *Theory of Voting*, Yale University Press, New Haven, CT, 1969.

FEIGENBAUM, J., PAPADIMITRIOU, C. H., and SHENKER, S., "Sharing the Cost of Multicast Transmissions," to appear in *J. Comput. System Sci.* [See also *Proceedings 32nd ACM Symposium on Theory of Computing*, (2000), 218–227.]

FELLER, W., *An Introduction to Probability Theory and Its Applications*, 3rd ed., Wiley, New York, 1968.

FERRETTI, V., NADEAU, J., and SANKOFF, D., "Original Synteny," *Proc. 7th Symp. Combinatorial Pattern Matching, Springer LNCS 1075* (1996), 149–167.

FINE, T. L., *Theories of Probability*, Academic Press, New York, 1973.

FUDENBERG, D., and TIROLE, J., *Game Theory*, MIT Press, Cambridge, MA, 1991.

GAMOW, G., "Possible Mathematical Relation between Deoxyribonucleic Acid and Proteins," *K. Dan. Vidensk. Selsk. Biol. Medd., 22* (1954), 1–13. (a)

GAMOW, G., "Possible Relations between Deoxyribonucleic Acid and Protein Structures," *Nature, 173* (1954), 318. (b)

GAREY, M. R., and JOHNSON, D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, 1979. [114]

GOLOMB, S. W., "Efficient Coding for the Deoxyribonucleic Channel," in *Mathematical Problems in the Biological Sciences*, Proceedings of Symposia in Applied Mathematics, Vol. 14, American Mathematical Society, Providence, RI, 1962, 87–100.

GRAHAM, R. L., *Rudiments of Ramsey Theory*, CBMS Regional Conference Series in Mathematics, No. 45, American Mathematical Society, Providence, RI, 1981.

GRAHAM, R. L., ROTHSCHILD, B. L., and SPENCER, J. H., *Ramsey Theory*, 2nd ed., Wiley, New York, 1990.

GRAVER, J. E., and YACKEL, J., "Some Graph Theoretic Results Associated with Ramsey's Theorem," *J. Comb. Theory, 4* (1968), 125–175.

GRAY, F., "Pulse Code Communication," U.S. Patent 2,632,058, March 17, 1953.

GREENWOOD, R. E., and GLEASON, A. M., "Combinatorial Relations and Chromatic Graphs," *Canadian J. Math., 7* (1955), 1–7.

GRIFFITHS, A. J. F., MILLER, J. H., SUZUKI, D. T., LEWONTIN, R. C., and GELBART, W. M., *An Introduction to Genetic Analysis*, 6th ed., Freeman, New York, 1996.

GRINSTEAD, C. M., and ROBERTS, S. M., "On the Ramsey Numbers $R(3,8)$ and $R(3,9)$," *J. Comb. Theory, Series B, 33* (1982), 27–51.

GUSFIELD, D., *Algorithms on Strings, Trees and Sequences; Computer Science and Computational Biology*, Cambridge University Press, New York, 1997.

HANNENHALLI, S., and PEVZNER, P. A., "To Cut or Not to Cut: Applications of Comparative Physical Maps in Molecular Evolution," *Proc. 7th ACM-SIAM Symp. Discrete Algs.* (1996), 304–313.

HARDY, G. H., and WRIGHT, E. M., *An Introduction to the Theory of Numbers*, 5th ed., Oxford University Press, New York, 1980.

HARRISON, M. A., *Introduction to Switching and Automata Theory*, McGraw-Hill, New York, 1965.

HARVARD COMPUTATION LABORATORY STAFF, *Synthesis of Electronic Computing and Control Circuits*, Harvard University Press, Cambridge, MA, 1951.

HEATH, L. S., and VERGARA, J. P. C., "Sorting by Bounded Block-Moves," *Discrete Appl. Math., 88* (1998), 181–206.

HERZOG, S., SHENKER, S., and ESTRIN, D., "Sharing the 'Cost' of Multicast Trees: An Axiomatic Analysis," *IEEE/ACM Trans. Networking, 5* (1997), 847–860.

HILL, F. J., and PETERSON, G. R., *Switching Theory and Logical Design*, Wiley, New York, 1968.

HOLLEY, R. W., EVERETT, G. A., MADISON, J. T., MARQUISEE, M., and ZAMIR, A., "Structure of a Ribonucleic Acid," *Science, 147* (1965), 1462–1465.

HOPCROFT, J. E., "Recent Developments in Random Algorithms," paper presented at SIAM National Meeting, Troy, NY, June 1981.

HUTCHINSON, G., "Evaluation of Polymer Sequence Fragment Data Using Graph Theory," *Bull. Math. Biophys., 31* (1969), 541–562.

HWANG, F. K., MONMA, C., and ROBERTS, F. S. (eds.), *Reliability of Computer and Communication Networks*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 5, American Mathematical Society and Association for Computing Machinery, Providence, RI, 1991.

JAIN, K., and VAZIRANI, V. V., "Applications of Approximation Algorithms to Cooperative Games," *Proceedings 33nd ACM Symposium on Theory of Computing*, (2001), 364–372.

JERRUM, M. R., "The Complexity of Finding Minimum-Length Generator Sequences," *Theoret. Comput. Sci., 36* (1985), 265–289.

115

JOHNSTON, R. J., "On the Measurement of Power: Some Reactions to Laver," *Environment and Planning A, 10* (1978), 907–914.

JONES, A. J., *Game Theory: Mathematical Models of Conflict*, Wiley, New York, 1980.

KALBFLEISCH, J. G., "Construction of Special Edge-Chromatic Graphs," *Canadian Math. Bull., 8* (1965), 575–584.

KALBFLEISCH, J. G., "Chromatic Graphs and Ramsey's Theorem," Ph.D. thesis, University of Waterloo, Waterloo, Ontario, Canada, January 1966.

KAPLAN, H., SHAMIR, R., and TARJAN, R. E., "Faster and Simpler Algorithms for Sorting Signed Permutations by Reversal," *Proc. 8th ACM-SIAM Symp. Discrete Algs.* (1997).

KARP, R. M., "Reducibility among Combinatorial Problems," in R. E. Miller and J. W. Thatcher (eds.), *Complexity of Computer Computations*, Plenum Press, New York, 1972, 85–103.

KARP, R. M., and LUBY, M. G., "Monte-Carlo Algorithms for Enumeration and Reliability Problems," *24th Annual ACM Symposium on Theory of Computing* (1983), 56–64.

KECECIOGLU, J. D., and SANKOFF, D., "Efficient Bounds for Oriented Chromosome Inversion Distance," *Proc. 5th Symp. Combinatorial Pattern Matching, Springer LNCS 807* (1994), 307–325.

KECECIOGLU, J. D., and SANKOFF, D., "Exact and Approximation Algorithms for Sorting by Reversal," *Algorithmica, 13* (1995), 180–210.

KELLY, J. S., *Social Choice Theory: An Introduction*, Springer-Verlag, New York, 1987.

KÉRY, G., "On a Theorem of Ramsey," *Matematikai Lapok, 15* (1964), 204–224.

KOHAVI, Z., *Switching and Finite Automata Theory*, McGraw-Hill, New York, 1970.

LANDER, E. S., and WATERMAN, M. S. (eds.), *Calculating the Secrets of Life; Contributions of the Mathematical Sciences to Molecular Biology*, National Academy Press, Washington, DC, 1995.

LAWLER, E. L., *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.

LEHMER, D. H., "The Machine Tools of Combinatorics," in E. F. Beckenbach (ed.), *Applied Combinatorial Mathematics*, Wiley, New York, 1964, 5–31.

LEWIN, B., *Genes VII*, Oxford University Press, New York, 2000.

LIU, C. L., *Introduction to Combinatorial Mathematics*, McGraw-Hill, New York, 1968.

LIU, C. L., *Elements of Discrete Mathematics*, McGraw-Hill, New York, 1977.

LUCAS, W. F., "Applications of Cooperative Games to Equitable Allocation," in W. F. Lucas (ed.), *Game Theory and Its Applications*, Proceedings of Symposia in Applied Mathematics, Vol. 24, American Mathematical Society, Providence, RI, 1981, 19–36. (a)

LUCAS, W. F. (ed.), *Game Theory and Its Applications*, Proceedings of Symposia in Applied Mathematics, Vol. 24, American Mathematical Society, Providence, RI, 1981. (b)

LUCAS, W. F., "Measuring Power in Weighted Voting Systems," in S. J. Brams, W. F. Lucas, and P. D. Straffin (eds.), *Political and Related Models*, Vol. 2 of *Modules in Applied Mathematics*, Springer-Verlag, New York, 1983, 183–238.

MAHADEV, N. V. R., and ROBERTS, F. S., "Consensus List Colorings of Graphs and

Physical Mapping of DNA," in M. Janowitz, F.-J. LaPointe, F. R. McMorris, B. Mirkin, and F. S. Roberts (eds.), *Bioconsensus*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 61, American Mathematical Society, Providence, RI, 2003, 83–95. [116]

MANN, I., and SHAPLEY, L. S., "Values of Large Games IV: Evaluating the Electoral College by Monte Carlo Techniques," RAND Corporation Memorandum RM-2651, September 1960; reproduced in M. Shubik (ed.), *Game Theory and Related Approaches to Social Behavior*, Wiley, New York, 1964. (a)

MANN, I., and SHAPLEY, L. S., " Values of Large Games Vl: Evaluating the Electoral College Exactly," RAND Corporation Memorandum RM-3158-PR, May 1962; reproduced in part in M. Shubik (ed.), *Game Theory and Related Approaches to Social Behavior*, Wiley, New York, 1964. (b)

MCKAY, B. D., and MIN, Z. K., "The Value of the Ramsey Number $R(3,8)$," *J. Graph Theory, 16* (1992), 99–105.

MCKAY, B. D., and RADZISZOWSKI, S. P., "$R(4,5) = 25$," *J. Graph Theory, 19* (1995), 309–322.

MOSIMANN, J. E., *Elementary Probability for the Biological Sciences*, Appleton-Century-Crofts, New York, 1968.

MOSIMANN, J. E., SHAPIRO, M. B., MERRIL, C. R., BRADLEY, D. F., and VINTON, J. E., "Reconstruction of Protein and Nucleic Acid Sequences IV: The Algebra of Free Monoids and the Fragmentation Stratagem," *Bull. Math. Biophys., 28* (1966), 235–260.

MUROGA, S., *Logic Design and Switching Theory*, Krieger Publishing Company, Melbourne, 1990.

MYERSON, R. B., *Game Theory*, Harvard University Press, Cambridge, MA, 1997.

NIVEN, I., *An Introduction to the Theory of Numbers*, Wiley, New York, 1991.

O'BRIEN, S. J. (ed), *Genetic Maps: Locus Maps of Complex Genomes*, 6th ed., Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, 1993.

OKADA, N., HASHIMOTO, T., and YOUNG, P., "Cost Allocation in Water Resources Development," *Water Resources Research, 18* (1982), 361–373.

OWEN, G., *Game Theory*, 3rd ed., Academic Press, San Diego, CA, 1995.

PALMER, J. D., OSORIO, B., and THOMPSON, W. F., "Evolutionary Significance of Inversions in Legume Chloroplast DNAs," *Current Genetics, 14* (1988), 65–74.

PARZEN, E., *Modern Probability Theory and Its Applications*, Wiley, New York, 1992.

PATTAVINA, A., *Switching Theory*, Wiley, New York, 1998.

PEVZNER, P. A., *Computational Molecular Biology: An Algorithmic Approach*, MIT Press, Cambridge, MA, 2000.

PRATHER, R. E., *Discrete Mathematical Structures for Computer Science*, Houghton Mifflin, Boston, 1976.

RADZISZOWSKI, S. P., "Small Ramsey Numbers," *Electron. J. Combinat., DS1* (2002), 35pp.

RAMSEY, F. P., "On a Problem of Formal Logic," *Proc. Lond. Math. Soc., 30* (1930), 264–286.

REINGOLD, E. M., NIEVERGELT, J., and DEO, N., *Combinatorial Algorithms: Theory and Practice*, Prentice Hall, Englewood Cliffs, NJ, 1977.

RIORDAN, J., *An Introduction to Combinatorial Analysis*, Princeton University Press, Princeton, NJ, 1980.

ROBERTS, F. S., *Discrete Mathematical Models, with Applications to Social, Biological, and Environmental Problems*, Prentice Hall, Englewood Cliffs, NJ, 1976.

ROBERTS, F. S., *Measurement Theory, with Applications to Decisionmaking, Utility, and the Social Sciences*, Addison-Wesley, Reading, MA, 1979.

ROBERTS, F. S., "Applications of Ramsey Theory," *Discrete Appl. Math., 9* (1984), 251–261. [117]

ROSS, S. M., *A First Course in Probability*, 5th ed., Prentice Hall, Upper Saddle River, 1997.

SESSIONS, S. K., "Chromosomes: Molecular Cytogenetics," in D. M. Hillis and C. Moritz (eds.), *Molecular Systematics*, Sinauer, Sunderland, MA, 1990, 156–204.

SETUBAL, J. C., and MEIDANIS, J., *Introduction to Computational Molecular Biology*, PWS Publishers, Boston, 1997.

SHAPLEY, L. S., "A Value for n-Person Games," in H. W. Kuhn and A. W. Tucker (eds.), *Contributions to the Theory of Games*, Vol. 2, Annals of Mathematics Studies No. 28, Princeton University Press, Princeton, NJ, 1953, 307–317.

SHAPLEY, L. S., "Measurement of Power in Political Systems," in W. F. Lucas (ed.), *Game Theory and Its Applications*, Proceedings of Symposia in Applied Mathematics, Vol. 24, American Mathematical Society, Providence, RI, 1981, 69–81.

SHAPLEY, L. S., and SHUBIK, M., "A Method for Evaluating the Distribution of Power in a Committee System," *Amer. Polit. Sci. Rev., 48* (1954), 787–792.

SHAPLEY, L. S., and SHUBIK, M., "On the Core of an Economic System with Externalities," *Amer. Econ. Rev., 59* (1969), 678–684.

SHUBIK, M., "Incentives, Decentralized Control, the Assignment of Joint Costs and Internal Pricing," *Management Sci., 8* (1962), 325–343.

STAHL, S., *A Gentle Introduction to Game Theory*, American Mathematical Society, Providence, RI, 1998.

STANAT, D. F., and MCALLISTER, D. F., *Discrete Mathematics in Computer Science*, Prentice Hall, Englewood Cliffs, NJ, 1977.

STONE, H. S., *Discrete Mathematical Structures and Their Applications*, Science Research Associates, Chicago, 1973.

STRAFFIN, P. D., JR., *Topics in the Theory of Voting*, Birkhäuser-Boston, Cambridge, MA, 1980.

STRAFFIN, P. D., and HEANEY, J. P., "Game Theory and the Tennessee Valley Authority," *Internat. J. Game Theory, 10* (1981), 35–43.

TVERSKY, A., and GILOVICH, T., "The Cold Facts about the 'Hot Hands' in Basketball," *Chance, 2* (1989), 16–21.

WATERMAN, M. S., *Introduction to Computational Biology; Maps, Sequences and Genomes*, CRC Press, Boca Raton, FL, 1995.

YOUNG, H. P., *Cost Allocation: Methods, Principles, Applications*, Elsevier Science, New York, 1986.

118

ZIMMER, C., "And One for the Road," *Discover*, January 1993, 91–92.

第3章 图论概述

3.1 基本概念[⊖]

3.1.1 一些例子

在例1.4中，我们简单地介绍了图的概念. 本章研究图和有向图及其在多方面的应用. 图是解决组合数学问题的基础工具. 而组合数学中的很多计数技术在解决图论问题中也很有用. 图论是一个古老的课题，近年来人们对它的兴趣与日俱增. 从一开始，这一课题就与应用密切相关. 它是由欧拉[1736]在解决著名的哥尼斯堡桥问题的过程中发明的，我们将在11.3.1节讨论这一著名的问题. 后来Kirchhoff[1847]把图论应用于电网的研究中，Cayley[1857，1874]把它应用于有机化学的研究中，哈密顿(Hamilton)把它应用于拼图游戏的研究中，而且很多数学家及非数学家把它应用于图和图着色的研究中. 20世纪，图论已日益广泛地应用于电机工程、计算机科学、化学、政治学、生态学、分子生物学、运输、信息处理和其他各个领域. 119

为了说明图论的应用，引出我们要介绍的图和有向图的形式定义，考虑下面几个例子.

【例3.1 运输网络】 图和有向图出现在很多运输问题中. 例如，考虑给定区域内的任意场所集合，在这些场所之间要求运输货物、人、汽车等. 这些场所可以是城市、仓库、街道拐角、飞机场以及类似的场所. 如图3.1a所示，把这些场所表示成点，而且当从场所 x 到场所 y 可以移送货物、人等时，从场所 x 到场所 y 画一箭头或有向直线(或曲线)，方向是从 x 到 y. 当所有连线都是双向时，我们可以用下面的方法更简单地把它们表示出来：在可以直接连接起来的两个场所之间画一条单一无向线，而不用对每对场所画出两个箭头［参见图3.1b］. 运输网络有诸多有趣问题，包括如何设计高效的交通网络，如何确保它们不易破坏，等等. ■

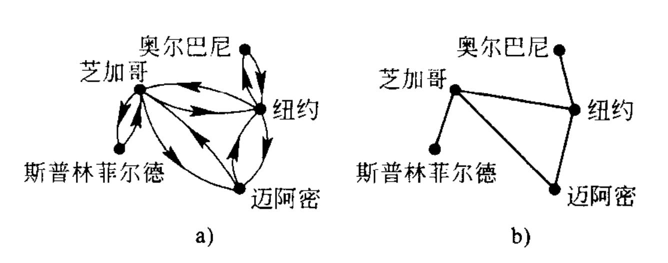


图3.1 有向航线

【例3.2 通信网络】 图也用于通信的研究中. 考虑一个委员会、一个法人团体或任意类似的需要联络的组织. 如图3.2所示，把这样的组织中的每一个成员表示成一个点，并且如果成员 x 能够与成员 y 直接联络，那么从成员 x 到成员 y 画一条有箭头的线. 例如，在图3.2的警察局中，长官能够直接与调度员联络，而这位调度员又能通过任意一位副官找到长官[⊜]. 关于这样的“通信网络”，所问及的典型问题类似于运输网络的问题：如何设计高效网络，破坏通信的容易程度等. 现代通信网络理论通常涉及交互通信网络和计算装备网络，而且它所生成的图是巨大的. ■ 120

⊖ 本章所介绍的图论的话题筛选有三个理由. 第一，它们生动地说明这一课题的性质和各方面应用. 第二，它们被用于说明了第2章介绍的计数技术. 第三，它们将用于说明第5章到第8章中的计数和存在结果. 我们将在第11章更全面地讨论图论，用连续三章讲述图和网络的内容，并开始介绍图论算法方面的内容. 在由Fred Roberts所执教的拉特格大学本科生组合数学的课程中，他没有涵盖太多图论的知识，因为有一门独立的本科生图论课程. 因此，他对本章的内容一带而过. 他分别讲授3.1.1节、3.1.2节和整个3.2节(但是只用大约30分钟，对练习也不做强调)以及3.3.1节、3.3.3节、3.4.1节、3.4.2节、3.5.1节、3.5.2节、3.5.4节、3.5.5节(只给出定理3.16的证明)和3.5.6节(不包括凯莱定理的证明概述)的相关内容. 可以加入其他各节扩展所涵盖的内容. 在图论课程及更多强调图、计算或理论等的组合数学课程中，应该包含比本章更多的内容.

⊜ 关于警察机构的类似的通信网络的更详细讨论可以参见Kemeny Snell[1962，Ch. 8].

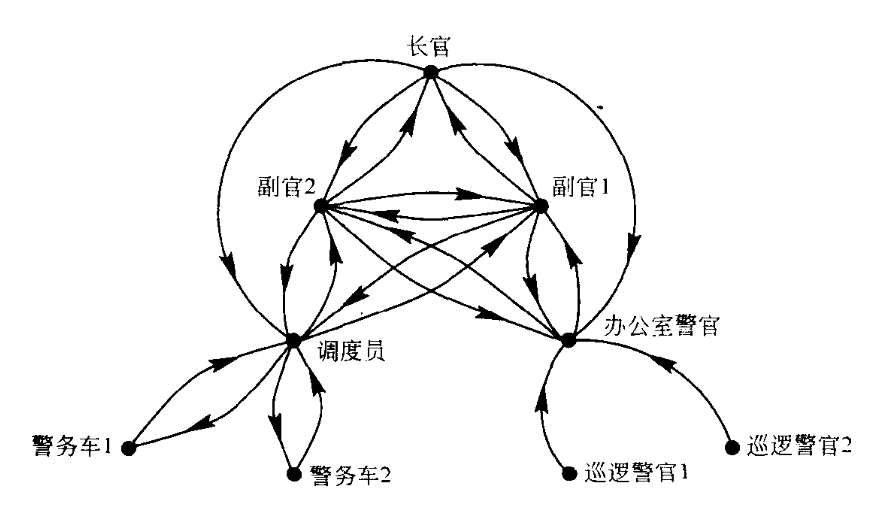


图 3.2　警察机构的部分通信网络

【例 3.3 物理网络】 图通常对应于物理系统．例如，可以把电网想象为一个图，点表示各电气部件，当且仅当存在一条连接两个点的电线时，用一条无向直线将它们相连．类似地，在电话网络中，我们可以把切换中心和各个电话想象成为点，且两个点有直线连接，如果它们之间存在直通电话线．例 1.3 所研究的原油和天然气管道也可以以这种方式翻译成图．我们通常寻找满足某种相互连接需求的最高效网络或成本最低的网络设计，或寻找最不易受到破坏的网络．■

【例 3.4 网络的可靠性】 如果运输、通信、电力或计算机网络被模型化成图，那么这一网络的组成部分的故障对应于点或线的故障．好的网络设计拥有冗余，这样，某些组成部分的故障不至于导致整个网络的故障(参见例 2.21)．例如，假设我们有如图 3.3所示由 6 台计算机组成的网络，且它们之间的直接连接表示成图中的直线．假设所有连接都存在一定的失效概率．当某些连接失效时，这一系统中的每一对计算机可以(至少间接地通过其他计算机)连通的概率是多少呢？能否把这一网络设计成为连接数相同但连通概率增加的不同网络呢？■

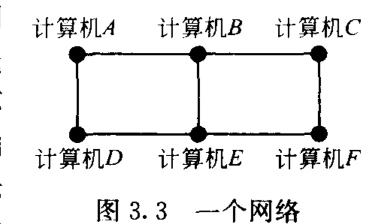


图 3.3　一个网络

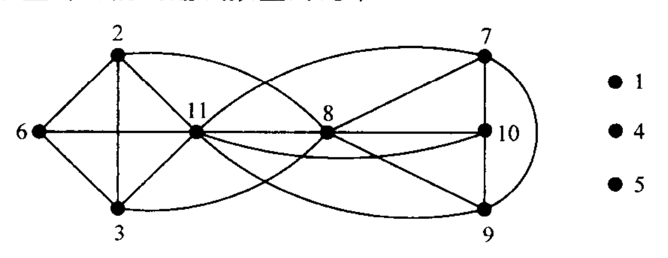
【例 3.5 因特网上的信息搜索】 由于因特网上的网站和网页数量的急剧增加，出现了诸如 Google 等的搜索引擎．这样的系统构建需要不断更新网页数据库．这种更新是通过周期性遍历网络的超文本结构的机器人实现的．假设把每个网站表示成一个点，并在其中从一个网站到另一个网站之间画箭头线，如果机器人可以在访问前一个网站之后访问后一个网站．应该以什么样的顺序访问各网站，使得最小化数据库中的网页过时的时间因素？答案部分依赖于给定网页的更新概率的评估．当我们查看用户查询的"最相关"网站或带足够高的"相关得分"的网站集合时，还要涉及因特网搜索中的图论的其他应用，这是第 12 章所讨论的匹配问题．■

121

【例 3.6 计算机程序分析】 在计算机程序分析中图有着非常广泛的应用．分析程序的方法之一是把大型程序细分成子程序，作为理解、文档制作或检查结构性错误的一种帮助．所考虑的子程序是程序块，或者是带有下面性质的计算机指令序列：一旦执行序列中的某个指令，那么序列中的所有指令都执行．设每个程序块被表示成点，如我们在图 3.1 中把城市表示成点一样．如果有可能把控制从程序块 x 的最后一个指令转换成程序块 y 的第一个指令，那么从 x 到 y 画一条有箭头的线．结果图表称为*程序有向图*(program digraph)(流程图是特殊的程序有向图，其中每个程序块具有一个指令)．某些程序块被设计成开始和结束块．为了检查程序中的错误，我们也许

可以使用编译器查询是否存在一些点(程序块)，从这些点出发，永远无法沿着箭头到达结束点. 或者我们也许可以使用编译器查询是否存在从开始点永远无法达到的点(如果是这样，这些无法达到的点对应于从不被调用的子程序). 程序有向图也可用于评估这一程序的运行时间. 图在计算机科学中有很多其他应用，如计算机和数字系统的设计与分析、数据结构、容错系统的设计以及数字系统的故障诊断等领域. ■

【例 3.7 物种间的竞争】 图也应用于生态系统的研究中. 考虑构成生态系统的若干物种. 把物种(或物种群)表示成点，如图 3.4 所示，并在物种 x 和 y 之间画一条无向线当且仅当 x 和 y 相互竞争. 结果图称为竞争图(competition graph，或生态位重叠图). 关于竞争图能提出的问题包括这些竞争图的结构性质的问题(例如它们如何连接，它们的连通部分是什么)，以及这些连线的密度问题(出现的连线数量与可能的连线数量的比率). ■ [122]



重要的物种：
1. 林冠：叶子，果实，花
2. 大的陆地动物：大的哺乳动物，鸟
3. 昆虫
4. 树干，果实，花
5. 陆地：根，落下的果实，叶子，树干
6. 真菌
7. 中部区域的攀爬动物：林冠中和陆地区域的哺乳动物
8. 林冠中的动物：鸟，果蝠，其他哺乳动物
9. 高空哺乳动物：鸟和蝙蝠，食虫的动物
10. 中部区域的飞行动物：鸟，食虫蝙蝠
11. 小的陆地动物：小的哺乳动物，鸟

图 3.4 马来群岛雨林的物种竞争图(数据来自于 Harrison[1962]，由 Cohen[1978]改编，图来自于 Roberts[1978])

【例 3.8 循环赛】 作为图的另一个应用，考虑网球循环赛㊀，其中每一名球手必须与其余每一位球手正好比赛一次，且不允许平局. 可以把球手表示成点并当"x 打败 y"时从 x 到 y 画一条有箭头的线，如图 3.5 所示. 类似的循环赛大量出现于社会学、生物学和环境科学中. 心理学家在一个选项集合上进行成对比较优先选择实验，对于这一选项集合中的每一对选项，要求对象说明他对此的喜好. 如果我们把这些选项对应于球手，而把"喜爱"对应于"打败"，那么这就定义一个循环赛. 生物学中也发生循环赛现象. 研究表明，农家的每一对小鸡中的一只"统治"另外一只. 这种小鸡间的啄食顺序同样定义一个循环赛(这同样也适合其他动物). 在循环赛的研究中，一个基础问题是确定"赢者"并排列"球手". 图论也有助于解决这一问题. ■

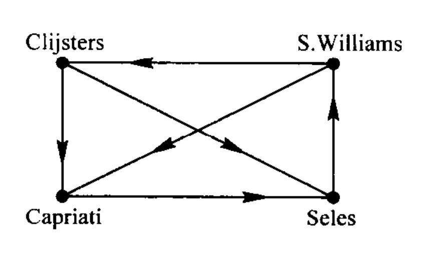


图 3.5 一个循环赛

【例 3.9 信息检索】㊁ 在计算机信息检索系统中，被索引的每一个文件被标上若干索引项(index term)或描述符(descriptor). 把这些索引项画成点，如果两个点所对应的索引项有密切关 [123]

㊀ 这不是一般的淘汰赛.

㊁ 这一例子来自于 Deo[1974].

系，那么在这两个点之间画一条线，如图 3.6 所示. 结果图称为类似图(similarity graph). 这一图可以用于生成文件分类，有助于信息检索：提供若干索引项，信息检索系统生成一个相关项和对应文件的列表. 在诸如万维网(练习 3.5)这样巨大的数据库的先进信息检索应用中，我们可能为特定的咨询项评估文件或网页的相关性，并寻找有最大化相关性的网页. ■

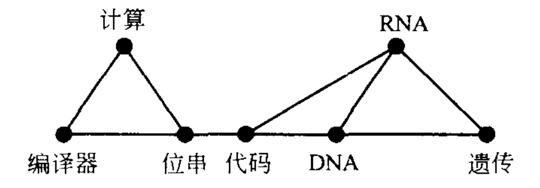


图 3.6　类似图的一部分

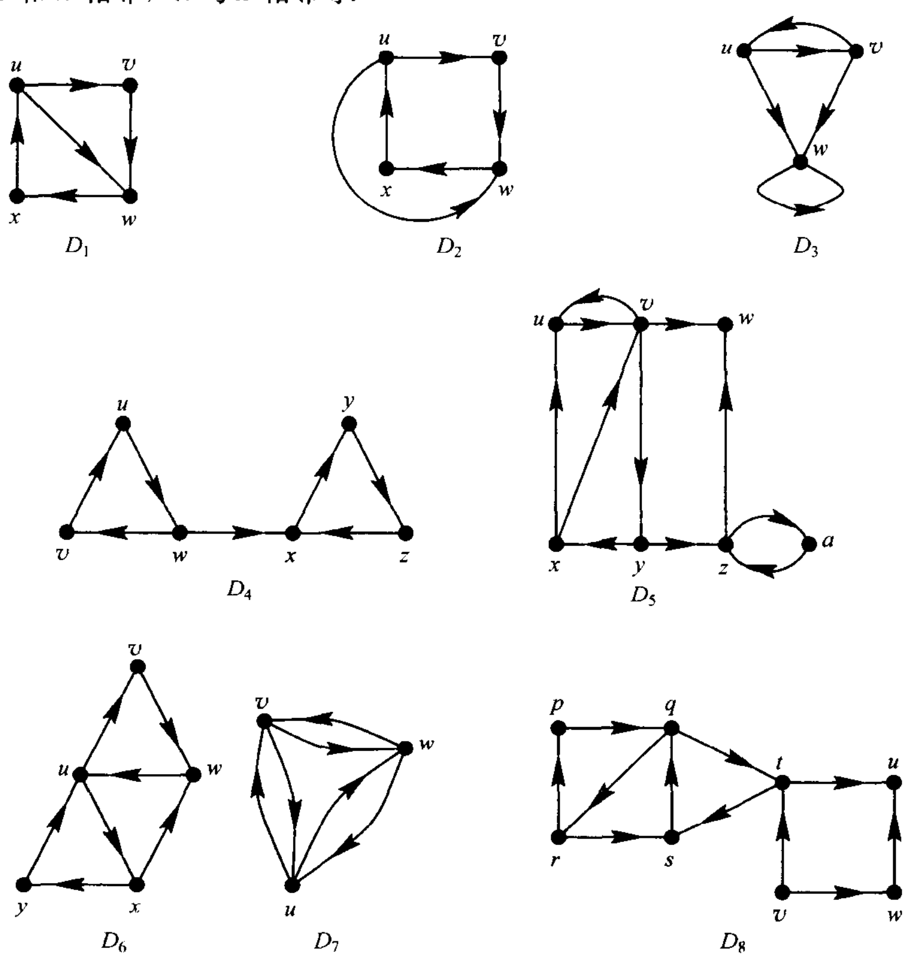
3.1.2　有向图和图的定义

这些例子都生成有向图或无向图. 我们更精确地将一个有方向的图或有向图 D(directed graph or digraph D)定义为一个序对(V, A)，其中 V 是一个非空集，而 A 是 V 中元素的序对的集合. V 被称为顶点(vertice)集合，而 A 是弧(arc)集合(有些作者使用术语节点(node)、点(point)等代替顶点，并用术语箭头(arrow)、有向线(directed line)、有向边(directed edge)或有向连接(directed link)代替弧). 如果要考虑多个有向图，那么我们将用$V(D)$和 $A(D)$分别表示 D 的顶点集合和弧集合. 通常，使用图 3.7 所示的那些图表示有向图. 这里，顶点被表示成点，而且存在从 u 到 v 的有向线(或者箭头，但没有必要是直的)当且仅当(u, v)在 A 中. 例如，对于图 3.7所示的有向图 D_1，V 是集合$\{u, v, w, x\}$且 A 是集合

$$\{(u,v),(u,w),(v,w),(w,x),(x,u)\}.$$

如果存在从顶点 u 到顶点 v 的弧，我们就说 u 与 v 相邻(adjacent). 因此，在图 3.7 的有向图 D_1 中，u 与 v 和 w 相邻，w 与 x 相邻等.

124



125

图 3.7　有向图

注意　读者应该注意到，在有向图的图表示中，顶点的放置位置不是重要的．顶点之间的距离也没有意义，连接顶点的线的性质也不重要等．另外，两个弧交叉与否也不重要；交叉点不是有向图的顶点．有向图中的所有信息就是观察给定的一对顶点是否由一条有向线或弧相连接以及弧的走向等．⊖ 因此，图 3.7 中的有向图 D_1 和 D_2 是相同的有向图，只是画法不同．在下面小节中，我们将说明 D_1 和 D_2 是同构的．

在有向图中，完全有可能在两个方向上都有弧，有从 u 到 v 的弧和从 v 到 u 的弧，如在图 3.7 的D_3 中所示的那样．也可能有从一个顶点到其自身的弧，如 D_3 中的顶点 w．这样的弧称为环(loop)．然而，从 u 到 v 的弧的数量却不可能大于 1．通常在有向图的理论和应用中，这样的多重弧是很有用的，例如在化学键的研究中就是如此．那时，我们需要研究多重图(multigraph)或多重有向图(multidigraph)，而不是有向图．

经常出现的情况是，当存在从 u 到 v 的弧时也存在从 v 到 u 的弧．在这种情况下，我们称有向图(V, A)是图(graph)．图 3.8 给出了几个图．在画图时，我们不考虑箭头，而且用连接顶点 u 和顶点 v 的单一无向线取代顶点 u 和顶点 v 之间的一对弧(并用无向环取代有向环)．我们称这样的线为这个图的一条边(edge)，并把它看成顶点的无序对$\{u, v\}$(顶点 u，v 可以相同)．如果图中存在边$\{u, v\}$，那么我们称 u 和 v 是邻居(neighbor)．以这种方式画出图 3.8 中的图，就得到图 3.9．因此，图 G 可以定义为对(V, E)，其中 V 是顶点集合，E 是 V 中无序元素对的集合，即边的集合．如果考虑多个图，我们将使用 $V(G)$和$E(G)$分别表示 G 的顶点集合和 G 的边的集合．

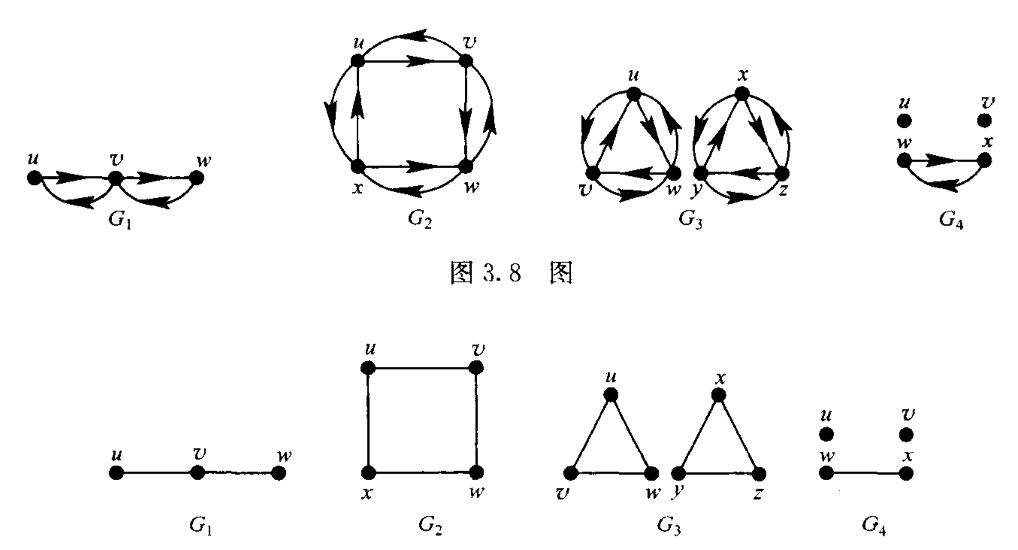


图 3.8　图

图 3.9　用边取代图 3.8 中的弧所得的图

这里，我们要明确有关图和有向图的几个假设．很多图论研究者都明确地做了下面的假设：不存在多重弧或边，即从顶点 u 到顶点 v 的弧或边的数量不大于 1．对于我们来说，这一假设已包含于有向图和图的定义当中．至少在最初我们将假设有向图和图没有环(我们对无环的有向图和图的几乎所有陈述对有环的有向图和图也成立)．我们也仅限于考虑带有有限个顶点的有向图和图．概括这些假设如下：

假设　除非特别指定，本书中所提到的所有有向图和图只有有限个顶点的集合，没有环，而

⊖ 这不完全是对的．有向图的某些性质有可能与图中顶点和弧的实际放置位置有关．我们将在下文中看到这一点．

且不允许有多重弧或边.

图 G 的顶点 u 的度(degree)$\deg(u)$或 $\deg_G(u)$计数 u 的邻居数目. 注意，如果我们把 G 的所有顶点的度加起来，那么我们就计数每一条边两次，计数每一个顶点一次. 因此我们有下面的定理.

定理 3.1 如果 G 是有 e 条边的任意图，那么有

$$\sum_{u \in V(G)} \deg(u) = 2e.$$

3.1.3 标签有向图和同构问题[⊖]

n 顶点标签有向图(labeled digraph)或 n 顶点标签图(labeled graph)是把整数 1，2，…，n 分别指定给每一个顶点的有向图或图. 实际上，两个标签有向图或图可以是一样的. 例如，图 3.10给出一个未标签图 G 和 G 的顶点的三个标签指定. 前两个标签指定在下面的意义下可以认为是相同的：它们的边集合是相同的. 然而，第一个和第三个标签指定是不同的，因为第一个指定有边{3，4}而第三个指定没有.

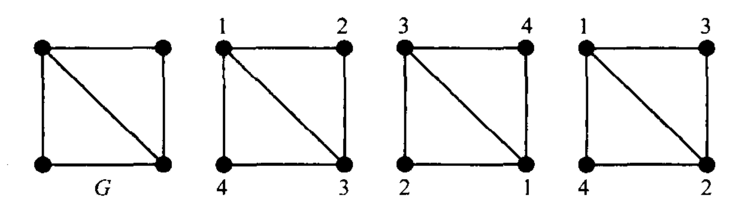


图 3.10 一个图 G 和它的顶点的三个标签指定

作为计数问题中的一个简单例子，考虑对于 $n \geqslant 2$，有多少个不同的 n 顶点标签图？我们观察到一个有 n 个顶点的标签图至多可有 $C(n, 2)$条边，因为 $C(n, 2)$是 n 个顶点的无序顶点对的数量. 因此，我们很容易得到答案. 假设这个图有 e 条边. 那么我们必须从 $C(n, 2)$条可能的边中选取 e 条边. 由此可知，有 n 个顶点和 e 条边的标签图的数量 $L(n, e)$由下式给出：

127

$$L(n,e) = C(C(n,2),e) = \binom{\binom{n}{2}}{e}. \tag{3.1}$$

因此，根据加法规则和等式(3.1)，有 n 个顶点的标签图的数量 $L(n)$由下式给出：

$$L(n) = \sum_{e=0}^{C(n,2)} L(n,e) = \sum_{e=0}^{C(n,2)} C(C(n,2),e). \tag{3.2}$$

例如，如果 $n=3$，那么 $L(3, 0)=1$，$L(3, 1)=3$，$L(3, 2)=3$，$L(3, 3)=1$，以及 $L(3)=8$. 图 3.11 给出有 3 个顶点的 8 个标签图.

注意，等式(3.2)表明不同的标签图的数量随 n 的增加而非常快速地增加. 为了明白这一点，注意如果 $r=C(n, 2)$，那么利用定理 2.8，有

$$L(n) = \sum_{e=0}^{r} C(r,e) = 2^r,$$

所以

$$L(n) = 2^{n(n-1)/2}. \tag{3.3}$$

⊖ 本小节是可选的. 把它放到这里是因为它是第 2 章的计算技术的一个良好应用，其中的概念偶尔要用到. 可以在需要时返回来看这些内容.

0条边	1条边	2条边	3条边
$L(3,0)=1$	$L(3,1)=3$	$L(3,2)=3$	$L(3,3)=1$

图 3.11　有 3 个顶点的 8 个标签图

由(3.3)式给出的数随着 n 成指数快速增长. 存在太多的图，以至于无法通过枚举列出所有图来回答大多数图论问题.

两个标签有向图被认为是相同的，如果它们的弧集合是相同的. 有多少个有 n 个顶点的标签有向图呢？因为任意弧是顶点的序对且没有环，根据乘法规则，存在 $n(n-1)$ 条可能弧. 有 n 个顶点和 a 条弧的标签有向图的数量 $M(n,\ a)$ 由下式给出：

$$M(n,a)=C(n(n-1),a)=\binom{n(n-1)}{a}. \tag{3.4}$$

再根据加法规则，有 n 个顶点的标签有向图的数量 $M(n)$ 由下式给出：

$$M(n)=\sum_{a=0}^{n(n-1)}M(n,a)=\sum_{a=0}^{n(n-1)}C(n(n-1),a). \tag{3.5}$$

例如，如果 $n=3$，那么 $M(3,\ 0)=1$，$M(3,\ 1)=6$，$M(3,\ 2)=15$，$M(3,\ 3)=20$，$M(3,\ 4)=15$，$M(3,\ 5)=6$，$M(3,\ 6)=1$，以及 $M(3)=64$. 图 3.12 给出有 3 个顶点和 4 条弧的 15 个标签有向图.

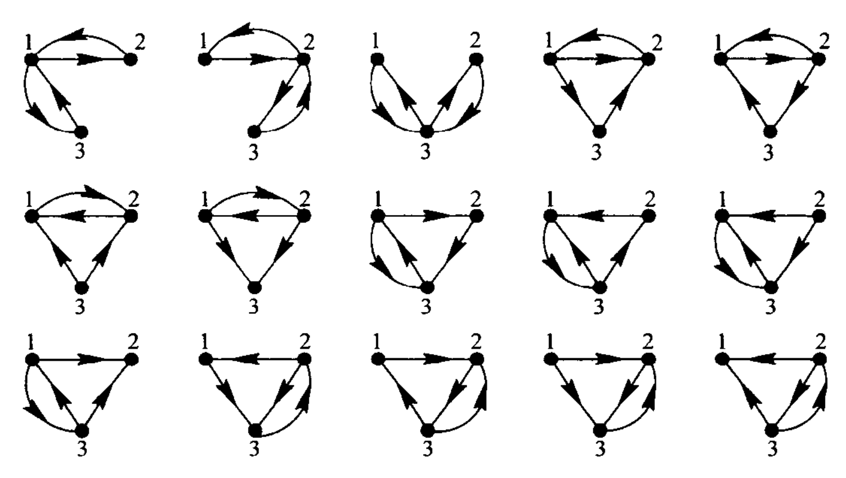


图 3.12　有 3 个顶点和 4 条弧的 15 个标签有向图

128 ~ 129

使用与上述关于 $L(n)$类似的论证，可以证明

$$M(n) = 2^{n(n-1)}. \tag{3.6}$$

等式(3.6)的证明留作练习.

两个有 n 个顶点的非标签图(有向图) G 和 H 被认为是相同的，如果二者的顶点可以用整数1，2，…，n 指定标签，使得边的集合(弧的集合)由相同的无序(序)对组成，即如果两个图(有向图)能够被标上标签，从而表明作为标签图(有向图)它们是相同的. 如果可以做到这一点，我们说 G 和 H 是**同构的**(isomorphic).

例如，图 3.13 中的图 G 和 H 是同构的，如图所示，标签指定为顶点 u 对应 1、v 对应 2、w 对应 3、x 对应 4、a 对应 1、b 对应 2、c 对应 3、d 对应 4. 图 3.7 中的有向图 D_1 和 D_2 也是同构的，在两个图中都把标签指定为 u 对应 1、v 对应 2、w 对应 3、x 对应 4 即可证明这一点.

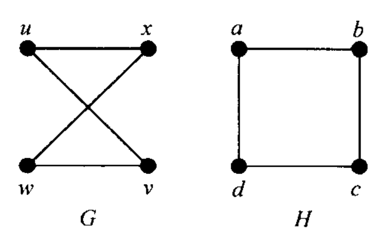


图 3.13 图 G 和 H 是同构的

尽管很容易区分两个标签图或标签有向图是否相同，但是确定两个未标签的图或有向图是否相同，即是否同构，的确是一个非常困难的问题. 这一问题称为**同构问题**(isomorphism problem)，是图论中最重要的问题之一. 确定有 n 个顶点的两个图 G 和 H 是否同构的大多数朴素算法使用整数 1，2，…，n 为 G 指定一个固定标签，然后使用这些整数尝试 H 的所有可能标签. 因此，这样的算法有计算复杂度 $f(n)=n!$，我们已在 2.3 节和 2.4 节中看到即使对于适当大小的 n，例如 $n=25$，考虑这样多的情况也是不可能的. 尽管已知一些好的算法，但解决同构问题的已知最好算法也有随问题大小成指数增长的复杂度，即对于适度大小的顶点数目，它们需要的计算步骤也不现实. 关于相关讨论可以参见 Reingold，Nievergelt 和 Deo[1977，Sec. 8. 5]、Deo[1974，Sec. 11-7]、Kreher 和 Stinson[1998]. 如果所讨论的图具有某些性质，则可用多项式算法. 例如，可参见 Luks[1982]、Bodlaender[1990]和 Ponomarenko[1984，1992].

130

本节练习

1. 对图 3.1 中的有向图，确定：
   (a)顶点集合 (b)弧的集合
2. 对于图 3.5 中的有向图重复练习 1.
3. 对于图 3.7 中的有向图 D_4 重复练习 1.
4. 对于图 3.9 中的每一个图，确定：
   (a)顶点集合 (b)边的集合
5. 对于图 3.7 中的有向图 D_5，寻找一个与顶点 y 邻接的顶点.
6. 对于图 3.1 中的图，寻找顶点纽约的所有邻居.
7. 画出一个运输网络，它以下面各城市作为顶点：纽约、巴黎、维也纳、华盛顿 DC、阿尔及尔，它有一条边连通两个城市，如果这两个城市之间可以通过陆路旅行.
8. 画出一个森林灭火队伍的通信网络.
9. 画出表示下面足球循环赛的有向图. 球队是：密歇根州队、俄亥俄州队和西北部球队. 密歇根州队胜俄亥俄州队、俄亥俄州队胜西北部球队、西北部球队胜密歇根州队.
10. 选择一个计算机程序并画出该程序的有向图.
11. 画出一个包含某些与生态学相关的术语的相似图.
12. 一个**食物网络**(food web)是一个有向图，它的顶点是一个生态系统中的某些物种，该图有从 x 到 y 的弧，如果 x 捕食 y. 画出下面这个物种集合的食物网络：{鹿，美洲狮，鹰，鼠，狐狸，草}.
13. 有时候，我们说在一个生态系统中的两个物种是竞争的，如果它们有一种共同的捕获物. 因此，我们可以使用食物网络构建一个竞争图(例 3.7). 给出练习 12 的食物网络的竞争图.
14. 扩展练习 13，我们定义对应于任意有向图 D 的竞争图 G：令 $V(G)=V(D)$ 且令 $\{x, y\}\in E(G)$ 当且仅当

存在 $a\in V(D)$，使得(x, a)和(y, a)在 $A(D)$中. 寻找对应于图 3.7 中的每一个有向图的竞争图.

15. 证明在带有 n 个顶点和 a 条边的图 G 中，存在一个度至少等于 $2e/n$ 的顶点.
16. 一个图的奇数度的顶点数量能是奇数吗？为什么？
17. 图 3.14 给出一个图和它的顶点的三个标签指定.

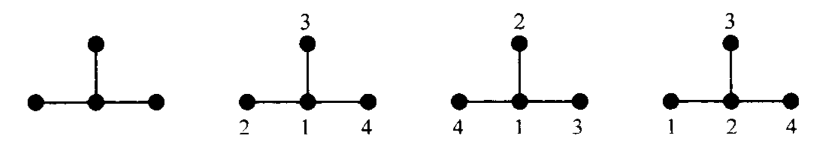


图 3.14 一个图和它的顶点的三个标签指定

(a)前两个标签图相同吗？为什么？

(b)第一个标签图和第三个标签图相同吗？为什么？

18. 图 3.15 给出一个有向图和它的顶点的三个标签指定.

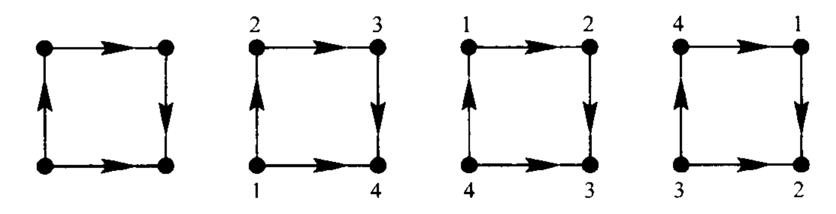


图 3.15 一个有向图和它的顶点的三个标签指定

(a)前两个标签图相同吗？为什么？

(b)第一个标签图和第三个标签图相同吗？为什么？ [131]

19. 使用等式(3.1)，给出有 4 个顶点和 2 条边的标签图的数量. 画出所有这样的标签图来检查你的答案.
20. 有 4 个顶点和偶数条边的标签图有多少个？
21. 使用等式(3.4)，寻找有 4 个顶点和 2 条弧的标签有向图的数量. 画出所有这样的标签有向图来检查你的答案.
22. 证明等式(3.6).
23. 图 3.16a 中的图是同构的吗？为什么？
24. 图 3.16b 中的图是同构的吗？为什么？

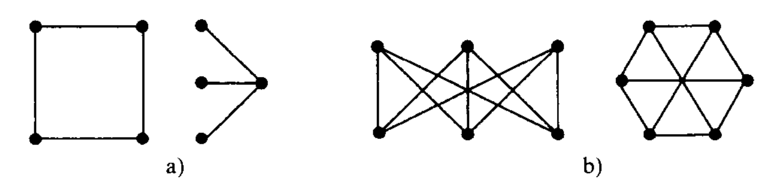


图 3.16 3.1 节的练习 23、24、28 的图

25. 图 3.17a 中的有向图是同构的吗？为什么？
26. 图 3.17b 中的有向图是同构的吗？为什么？
27. 图 3.17c 中的有向图是同构的吗？为什么？

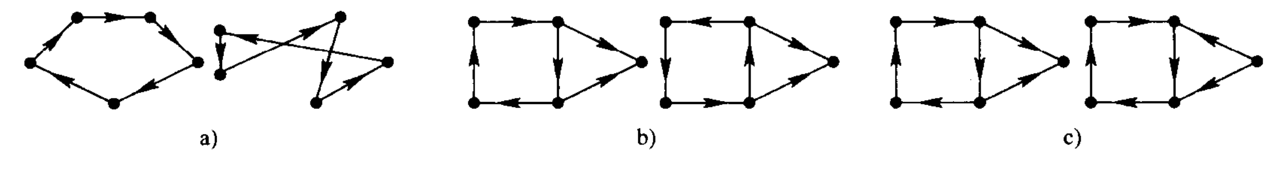


图 3.17 3.1 节的练习 25、26、27 的有向图

28. 通过用(x, y)或(y, x)中的一条弧取代图中的每一条边$\{x, y\}$，可以产生一个图的定向(orientation)。例如，图 3.15 中的有向图是图 3.13 中的图 H 的一个定向. 对图 3.16a 中的每一个图，寻找其所有非同构的定向.

132

29. 假设 G 和 H 是有着相同顶点数和相同边数的两个图. 假设 α_k 是 G 中正好有 k 个邻居的顶点数，且 β_k 是 H 中正好有 k 个邻居的顶点数. 假设对于所有 k，有 $\alpha_k=\beta_k$. G 和 H 一定同构吗？为什么？
30. 如果 $\alpha_2=\beta_2=|V(G)|=|V(H)|$，且对于 $k\neq 2$ 有 $\alpha_k=\beta_k=0$. 重复练习 29.

3.2 连通性

3.2.1 有向图中的可达性

在通信网络中，一个很自然的问题是：一个人能够把一条信息传送给另外一个人吗？在运输网络中，一个类似的问题是：一辆汽车能从位置 u 移动到位置 v 吗？在程序有向图中，我们感兴趣的是，确定是否可以从每一个顶点出发沿着弧移动并最终到达一个表示停止的顶点. 所有这些问题都与有向图 $D=(V, A)$中的可达性的思路一致：我们能否从顶点 u 出发，并沿着 D 中的弧到达顶点 v？

为了使这一概念更精确，引入一些定义. D 中的一条**路径**(path)是一个序列：

133

$$u_1, a_1, u_2, a_2, \cdots, u_t, a_t, u_{t+1}, \tag{3.7}$$

其中 $t\geqslant 0$，每一个 u_i 在 V 中，即 u_i 是一个顶点，且每一个 a_i 在 A 中，即它是一条弧，且 a_i 是弧(u_i, u_{i+1}). 即弧 a_i 从 u_i 到 u_{i+1}. 因为 t 可能是 0，所以 u_1 是一条路径，一条从 u_1 到 u_1 的路径. 如果我们使用相同顶点从不超过一次[⊖]，路径(3.7)称为**简单路径**(simple path). 例如，在图 3.7 的有向图 D_5 中，u，(u, v)，v，(v, w)，w 是简单路径，而 u，(u, v)，v，(v, y)，y，(y, x)，x，(x, v)，v，(v, w)，w 是非简单路径的路径，因为它使用顶点 v 两次. 当提到路径时，命名弧是多余的，所以我们只简单说(3.7)式为路径 $u_1, u_2, \cdots, u_t, u_{t+1}$. 可达性和连通性的对比如表 3.1 所示.

表 3.1 可达性和连通性

有向图 D $u_1, a_1, u_2, a_2, \cdots, u_t, a_t, u_{t+1}$ 可达性	图 G $u_1, e_1, u_2, e_2, \cdots, u_t, e_t, u_{t+1}$ 连通性
路径： a_i 是(u_i, u_{i+1})	**链：** e_i 是$\{u_i, u_{i+1}\}$
简单路径： 路径且 u_i 不同	**简单链：** 链且 u_i 不同
闭路径： 路径且 $u_{t+1}=u_1$	**闭链：** 链且 $u_{t+1}=u_1$
循环(简单闭路径)： 路径且 $u_{t+1}=u_1$， 对 $i\leqslant t$，u_i 各不相同，且(a_i 各不相同)[①]	**回路(简单闭链)：** 链且 $u_{t+1}=u_1$， 对 $i\leqslant t$，u_i 各不相同且 e_i 各不相同

① 从对 $i\leqslant t$，u_i 各不相同可得.

一条路径(3.7)式称为是**闭的**(closed)，如果 $u_{t+1}=u_1$. 闭路径结束于开始点. 如果路径(3.7)式是闭的且顶点 $u_1, u_2, \cdots, u_t$ 各不相同，那么(3.7)式称为**循环**(cycle，简单闭路

⊖ 图论学习中的一个难题是最初必须掌握大量的术语. 为了帮助读者克服这一困难，我们在表 3.1 中以简单的形式引入路径、简单路径等概念.

径[㊀]).(读者应该注意到如果对于 $i\leqslant t$ 顶点 u_i 是各不相同的，那么弧 a_i 也必是各不相同的.)

作为例子，图 3.7 中的有向图 D_1 的路径 u，v，w，x，y，u 是一个循环，有向图 D_6 中的路径 u，v，w，x，y，u 也是循环. 但是 D_5 中的闭路径 u，v，y，x，v，y，x，u 不是循环，因为存在重复的顶点. 一般地，在计数或列出有向图的循环时，我们将不区分以相同的顺序，使用相同的顶点和弧，但开始于不同顶点的循环. 因此，在图 3.7 的有向图 D_6 中，我们认为循环 w，x，y，u，v，w 与循环 u，v，w，x，y，u 相同. 路径、简单路径、循环等的长度(length)是这一路径中的弧的数量. 因此，路径(3.7)式有长度 t. 在图 3.7 的有向图 D_5 中，u，v，y，x，v 是一条长度为 4 的路径，而 u，v，y，z 是长度为 3 的简单路径，u，v，y，x，u 是长度为 4 的循环. 我们说 v 是从 u 开始可达的(reachable)，如果存在一条从 u 到 v 的路径. 因此，在 D_5，z 是从 u 开始可达的. 然而 u 不是从 z 开始可达的.

134

有向图 $D=(V, A)$称为是完全对称的(complete symmetric)，如果对于 V 中的每一个 $u\neq v$，序对(u, v)是 D 的一条弧. 例如，图 3.7 中的有向图 D_7 是完全对称的. n 个顶点上的完全对称有向图有 $n(n-1)$条弧. 设对于给定的 k，k 不超过图中的顶点数目 n，有多少条长度为给定长度 k 的简单路径? 答案就是寻找这样的一条路径：选任意 $k+1$ 个顶点，然后排列它们. 因此我们有 $C(n, k+1)\cdot(k+1)!=P(n, k+1)$条这样的路径.

3.2.2 图中的连通性

假设 $G=(V, E)$是一个图. 引入类似于有向图中的术语. G 中的链是一个序列：

$$u_1, e_1, u_2, e_2, \cdots, u_t, e_t, u_{t+1}, \tag{3.8}$$

其中 $t\geqslant 0$，每一个 u_i 是一个顶点，每一个 e_i 是边$\{u_i, u_{i+1}\}$. 一条链称为是简单的，如果所有的 u_i 不同；一条链称为是闭的，如果 $u_{t+1}=u_1$. u_1，u_2，…，u_t 各不相同且 e_1，e_2，…，e_t 各不相同时，闭链(3.8)称为回路(简单闭链[㊁]). 形如(3.8)式的链、回路等的长度是其中的边的数量. 我们说 u 和 v 是连通的，如果存在从 u 到 v 的链.

作为例子，在图 3.18 的图中，r，$\{r, t\}$，t，$\{t, w\}$，w，$\{w, u\}$，u，$\{u, t\}$，t，$\{t, r\}$，r，$\{r, s\}$，s，$\{s, u\}$，u，$\{u, t\}$，t，$\{t, w\}$，w 是一条链. 如果不考虑边这条链可以写成 r，t，w，u，t，r，s，u，t，w. 一个简单链由 r，t，u，w，x 给出. 一个回路由 r，t，u，s，r 给出. 最后，p，$\{p, r\}$，r，$\{r, p\}$，p 不是一条回路，因为 $e_1=e_t$.(边是无序的，所以$\{p, r\}=\{r, p\}$.)注意，如果没有边各不相同的限制，这是一个回路. 在与有向图类似的情况下，对于一个循环来说，我们没有假设弧是不同的，因为这可以从顶点各不相同得到.

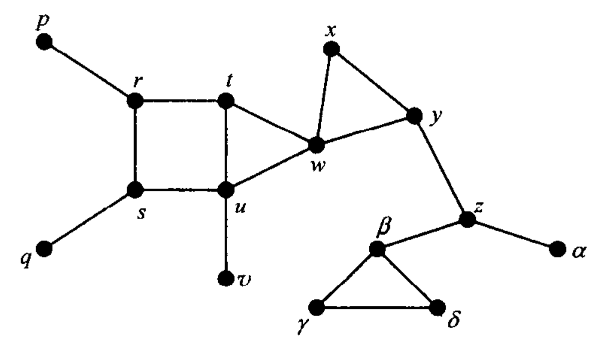


图 3.18 一个图

n 个顶点的完全图(complete graph)K_n 定义为每一对顶点都被一条边连通的图. 在 K_n 中，存在 $C(n, 2)$条边. K_n 中长度为 k 的简单链的数量由 $P(n, k+1)$给出.(为什么?)

㊀ 严格地说，简单闭路径不是简单路径.

㊁ 在下面我们将看到为什么要加入边各不相同的限制. 也应该注意到，严格说来，简单闭链不是简单的.

3.2.3　强连通有向图和连通图

图论很有用的原因之一是它的几何形式允许我们定义各种结构概念. 其中一个概念就是连通性. 一个有向图称为是**强连通的**(strongly connected), 如果对于每一对顶点 u 和 v, v 是从 u 可达的, 且 u 是从 v 可达的. 因此, 图 3.7 中的有向图 D_6 是强连通的, 但有向图 D_5 不是强连通的. 如果一个通信网络是强连通的, 那么每一个人都可以与另一个人通信. 如果一条运输网络不是强连通的, 那么存在两个位置 u 和 v 使得你不能从第一个位置走到第二个位置, 或者不能从第二个位置走到第一个位置. 在 11.2 节中, 我们将研究如何得到强连通运输网络. 一个程序有向图从不是强连通的, 因为不存在从停止顶点出发的弧. 一个循环赛可以是强连通的(参见图 3.5); 然而, 它不必一定是强连通的. 在一个强连通的循环赛中, 很难把球手分出等级. 因为我们会得到这样的局面, 其中 u_1 打败 u_2, u_2 打败 u_3, …, u_{t-1} 打败 u_t, u_t 打败 u_1.

[135]

我们说一个图是**连通的**(connected), 如果每一对顶点 u 和 v 之间存在一条链. 这一连通性的概念与拓扑学中所使用的概念是一致的: 这个图有一个"部分". 在图 3.9 中, 图 G_1 和 G_2 是连通的而 G_3 和 G_4 则不是连通的. 物理网络(电网、电话网、管道网络)通常是连通的. 事实上, 我们尝试构建这些物理网络, 使得在某一条边上的破损不会导致一个非连通图.

有用各种方法设计的检测一个图是否是连通图的算法. 最快的算法已经相当好. 它们的计算复杂度与顶点 n 的数量加上边 e 的数量是线性的. 使用 2.18 节中的语言, 它们的步骤数量级是 $n+e$(它们是 $O(n+e)$). 因为图至多有 $\binom{n}{2}$条边, 所以我们有

$$e\leqslant\binom{n}{2}=\frac{n(n-1)}{2}\leqslant n^2.$$

因此, $n+e\leqslant n+n^2$, 所以这些算法的数量级是 $n+n^2$, 这是一个 n 的多项式. (以 2.18 节的记法, 它们是 $O(n^2)$, 我们说它们是 n 的二次算法). 类似的结果和算法适用于有向图和强连通性的概念. 我们将在 11.1 节中讨论其中的一些算法. 参见 Aho, Hopcroft 和 Ullman[1974]、Even[1979]、Frank[1995]、Gibbons[1985]、Golumbic[1980]、Reingold, Nievergelt 和 Deo[1977]及 West[2001].

[136]

3.2.4　子图

在下文中有时候观察一个图的子图是很有用的. 假设 $G=(V, E)$是一个图. 子图(subgraph) $H=(W, F)$使得 W 是 V 的子集, F 是 E 的一个子集且是 W 的顶点的无序对的集合. 因此, 为了定义 G 的子图, 我们从 G 中选出一些顶点和连接所选出的顶点的一些边. 例如, 在图 3.19 中, 图 H 和 H'是图 G 的两个子图. 在 H'中, 边的集合是由连接 $W=\{a, c, d\}$顶点的所有 G 的边组成的. 在这样的情况下, 我们说 H'是由 W 的顶点**生成的子图**(subgraph generated)或**诱导**(induced)的子图.

类似的概念适用于有向图. 如果 $D=(V, A)$是一个有向图, 那么 D 的子图(或有向子图) $J=(W, B)$是如下有向图: W 是 V 的子集, B 是 A 的子集且是 W 的顶点的序对的子集. 如果 B 是连接 W 内的顶点的 D 的所有弧, 那么 J 是一个生成子图. 例如, 在图 3.20 中, 有向图 J 是有向图 D 的子图且有向图 J'是由顶点 a, c, d 生成的子图.

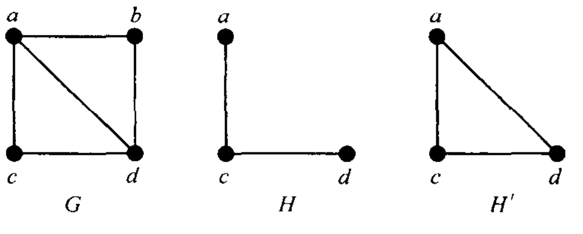


图 3.19　H 是 G 的一个子图, H'是一个生成子图

作为这些概念的一个简单应用, 考虑有 n 个

顶点的完全对称有向图存在多少个 k 个顶点的子图？为了寻找这样的子图，首先我们选出 k 个顶点；实现这一点有 $C(n,k)$ 种方法. 这些顶点由 D 内的 $k(k-1)$ 条弧连接. 我们可以为这个子图选择这一弧集合的任意子集；即我们可以有 $2^{k(k-1)}$ 种方法为这个子图选择弧. 因此，根据乘法规则，存在

$$C(n,k)\cdot 2^{k(k-1)}$$

个 k 顶点子图.

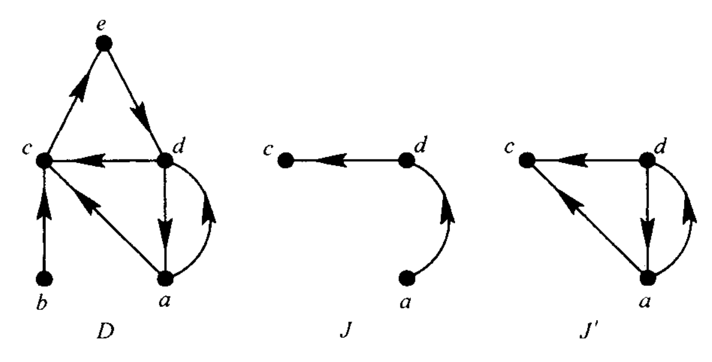


图 3.20 J 是 D 的一个子图，J' 是由顶点 a，c 和 d 生成的子图

137

【例 3.10 系统的可靠性(再探例 2.21)】 在例 2.21 中，我们研究了由可能运转也可能失效的组成部分组成的系统，并介绍了已知一些组成部分正常运转确定系统是否运转的规则. 在研究这样的系统可靠性时，我们一般用图 G 表示一个系统，并让每一条边对应一个组成部分. 于是在很多情况(参见例 3.4)下，我们说这一系统是运转的，当且仅当每一对顶点由运转组成部分的链连接，即当且仅当由 G 的所有顶点和运转边组成的子图 H 是连通的. 例如，考虑图 3.21a 的图 G. 存在三个组成部分，标记为 x_1，x_2 和 x_3. 显然，这一系统运转当且仅当至少有两个组成部分运转. 类似地，如果 G 如图 3.21b 所示，那么这一系统运转当且仅当成分 x_1 运转且三个剩余组成部分中至少有两个运转. ■

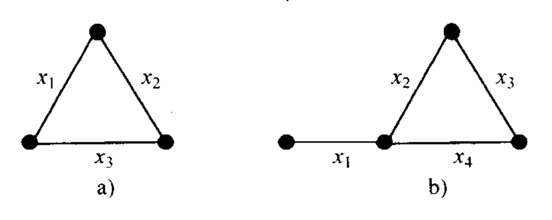


图 3.21 两个系统

3.2.5 连通分支

假设 $G=(V,E)$ 是一个图，G 的一个**连通分支**(connected component)或一个**分支**(component)是 G 的一个连通的生成子图 H，而且它在下面的意义下是最大的：不存在包含 H 的所有顶点的更大的 G 的连通生成子图 K. 例如，在图 3.22 的图 G 中，由顶点 a，b 和 c 生成的子图是连通的，但是它不是一个分支，因为由顶点 a，b，c 和 d 生成的子图是一个连通生成子图，它包含第一子图的所有顶点. 这第二个子图是一个分支. G 有三个分支，其他两个分支分别是由顶点 e，f 和 g 生成的子图和由 h 和 i 生成的子图. 这些分支对应于这一图中的“部分”. 显然，连通图正好有一个分支. 例如，在例 3.9 的信息检索中，分支产生文件的一个自然分类. 在图 3.4 的竞争图中，有四个分支(三个分支是由一个顶点组成的). 现实世界的竞争图倾向于至少有两个分支. 对于有向图中类似于连通分支的概念将在练习中加以讨论.

138

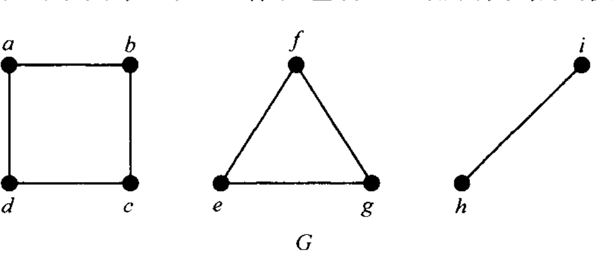


图 3.22 图 G 有三个分支，由顶点 a，b，c 和 d 生成的子图、由顶点 e，f 和 g 生成的子图和由顶点 h 和 i 生成的子图

本节练习

1. 对于图 3.7 中的有向图 D_8：
   (a)寻找一条非简单路径的路径.
   (b)寻找一条封闭路径.
   (c)寻找一条长度为 4 的简单路径.

(d)确定 q,(q, t),t,(t, s),s,(s, q),q 是否是一个循环.

(e)寻找一条包含顶点 p 且长度为 3 的循环.

2. 对于图 3.18 中的图:

(a)寻找一条不是回路的闭链.

(b)寻找最长的回路.

(c)寻找一条不同于文中给出的非简单链.

(d)寻找长度为 6 的闭链.

3. 给出一个有向图和其中的一条不是简单路径但没有重复弧的路径例子.

139

4. 给出一个图的例子,在这个图中最短的回路长度为 5 且最长的回路长度为 8.

5. 对于图 3.7 中的每一个有向图,确定它是否是强连通.

6. 图 3.23 中哪些图是连通的?

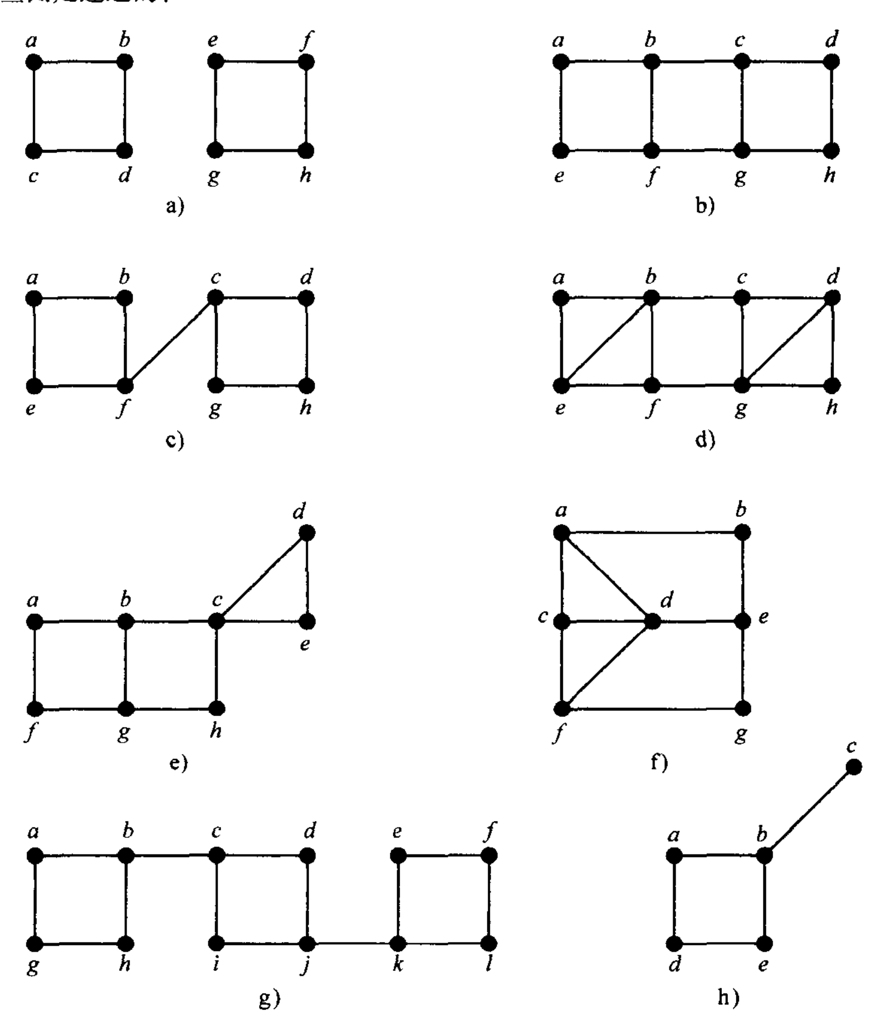


图 3.23 3.2 节练习中的图

7. 对于图 3.24 中的每一个有向图:

(a)寻找一个不是生成子图的子图.

(b)寻找由顶点 5,8 和 9 所生成的子图.

(c)寻找一个强连通生成子图.

8. 对于图 3.25 中的图:

(a)寻找一个不是生成子图的子图.

(b)寻找连通但不是连通分支的生成子图.

(c)寻找所有连通分支.

9. 一个有向图是**单连通的**(unilaterally connected)，如果对于每一对顶点 u 和 v，或者 v 是从 u 可达的或者 u 是从 v 可达的. 但不必二者同时成立.

(a)给出一个单连通的但不是强连通的有向图的例子.

(b)对于图 3.7 中的每一个有向图，确定它是否是单连通的.

10. 一个有向图是**弱连通的**(weakly connected)，如果不计所有弧的方向时，其结果图(或可能的多重图)是连通的.

(a)给出一个弱连通但不是单连通的有向图的例子.

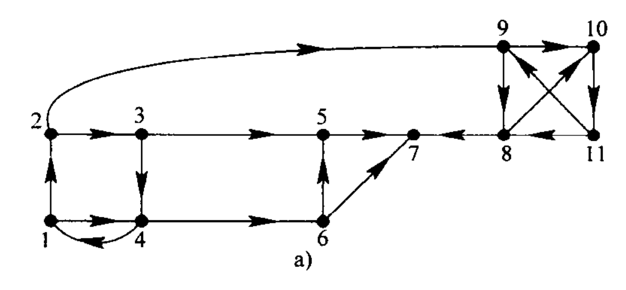
(b)给出一个非弱连通的有向图的例子.

(c)对于图 3.7 中的每一个有向图，确定它是否是弱连通的.

11. 证明：如果在有向图 D 中 v 是从 u 可达的，那么在 D 中存在一条从 u 到 v 的简单路径.

12. 假设一个由图 G 定义的系统运转当且仅当 G 的顶点和运转边形成 G 的一个连通子图. 在什么样的环境下图 3.26 所给定的每一个系统工作?

13. 在有向图 D 中，一个**强分支**(strong component)是一个强连通生成子图，且在下面的意义下它是最大的：它不被包含在任意更大的强连通生成子图中. 例如，在图 3.7 的有向图 D_5 中，由顶点 x，y 和 v 所生成的子图是强连通的但不是一个强分支，因为由顶点 x，y，v 和 u 所生成的子图也是强连通的. 而后者是一个强分支. 由单一顶点 w 所生成的子图也是强分支，且由顶点 z 和 a 所生成的子图也是强分支. 不存在其他强分支(对于把强分支应用于通信网络、能量需求和概率现象的马尔可夫模型等内容，可参见 Roberts[1976]). 寻找图 3.7 中每一个有向图的强分支.



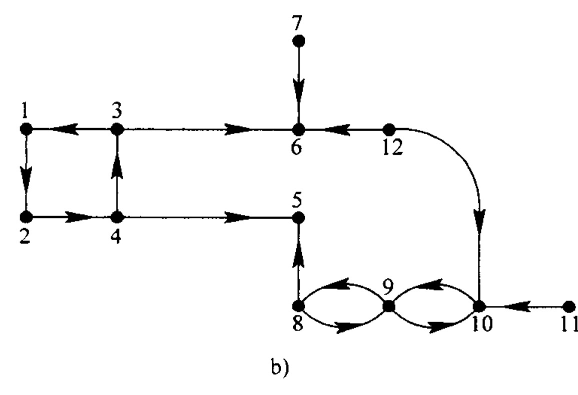


图 3.24　3.2 节练习中的图

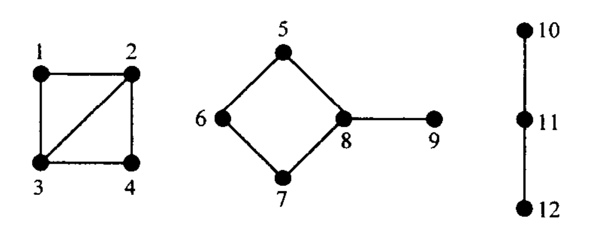
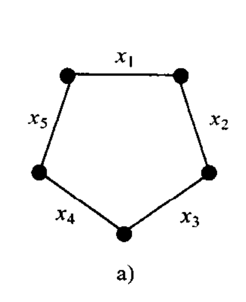
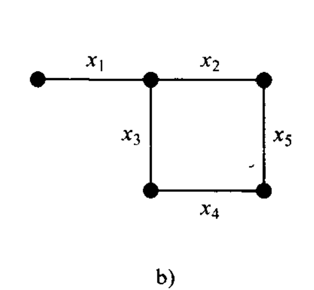


图 3.25　3.2 节练习中的图

140~142





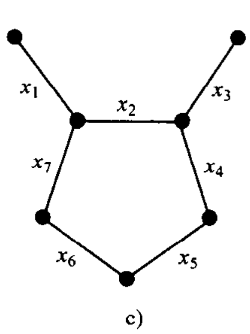


图 3.26　3.2 节练习 12 的系统

14. 寻找图 3.2 中的警察机构的所有强分支.

15. 在有向图 D 中，证明：

(a)每一个顶点都在某个强分支中.

(b)每一个顶点至多在一个强分支中.

16. 证明：一个图是连通的当且仅当它有一条通过所有顶点的链.
17. 证明：一个有向图是强连通的当且仅当它有一条通过所有顶点的封闭路径.
18. 证明：在一个单连通有向图 D 中，在任意顶点集合内，存在一个顶点，它可以达到(使用 D 的弧)这一集合中的其他所有顶点.
19. 在一个通信网络中，假设一条从 x 到 y 的弧意味着信息可以直接从 x 送到 y. 如果我们想用尽可能小的顶点集合放置信息，使得这条信息可以到达其他所有顶点(有可能通过多步)，当这个通信网络是下列各情况时，所需要的最少顶点数目是多少?
    (a)强连通.
    (b)单连通.
20. 在练习 19 中，如果这个通信网络是弱连通的，你能够总用至多一半的顶点放置信息并确保它能到达其他所有顶点吗?
21. 根据练习 18 的结果，证明一个有向图是单连通的，当且仅当它有一条通过所有顶点的路径.
22. (a)给出一个没有通过所有顶点的循环的强连通有向图的例子.
    (b)每一个单连通有向图都有一条通过所有顶点的路径吗?
23. 有向图的一个*弱分支*(weak component)是一个极大的弱连通生成子图. 对于图 3.24 中的每一个有向图，寻找所有弱分支.

143

24. 有向图的*单分支*(unilateral component)是一个极大的单连通的生成子图.
    (a)在图 3.24b 的有向图中寻找带有五个顶点的单分支.
    (b)有向图的每一个顶点至少在一个单分支中吗?
    (c)每一个顶点能在多个单分支中吗?
25. 一个有向图是*单路径的*(unipathic)，如果 v 是从 u 可达的，且正好存在一条从 u 到 v 的简单路径.
    (a)图 3.7 中的有向图 D_4 是单路径的吗?
    (b)图 3.15 中的有向图又如何呢?
26. 对于强连通且有 n 个顶点的有向图，弧的最少数量是多少? 最多数量又是多少?(注意，强连通且有最少数量弧的有向图很容易被干扰. 为了干扰通信必须切断多少链接?)
27. (Harary，Norman 和 Cartwright[1965])考虑练习 25 中提到单路径的定义. 一个单路径有向图的两个循环能够有公共弧吗?(给出证明或举反例.)
28. (Harary，Norman 和 Cartwright[1965])如果 D 是强连通的且至少有两个顶点，这时，每一个顶点都必须在一个循环上吗?(给出证明或举反例.)
29. 假设有向图 D 不是弱连通的.
    (a)如果 D 有四个顶点，弧的最大数量是多少?
    (b)如果 D 有 n 个顶点，又如何?
30. 对于单连通但不强连通的有向图做练习 29.
31. 对于弱连通但不单连通的有向图做练习 29.
32. 当单一弧或顶点失效时模型化为有向图 D 的网络的可靠性可以通过它的连通性的改变量来测定. 设 $D-u$ 是由顶点 u 之外的顶点生成的子图. 给出一个有向图 D 的例子，它的顶点 u 带有下列性质；或证明不存在这样的有向图：
    (a)D 是强连通的且 $D-u$ 是单连通但不是强连通的.
    (b)D 是强连通的且 $D-u$ 不是单连通的.
    (c)D 是单连通但不是强连通的且 $D-u$ 不是单连通的.
33. 设 a 是 D 的一条弧，且 $D-a$ 是移去弧 a 所得的 D 的子图. 对 $D-a$ 重复练习 32.
34. (Harary，Norman 和 Cartwright[1965])如果 D 是一个有向图，定义*补有向图* D^c(complementary digraph)如下：$V(D^c)=V(D)=V$ 且 $V\times V$ 的序对(u, v)(且 $u\neq v$)在 $A(D^c)$中，当且仅当它不在 $A(D)$

内. 例如，如果D是图3.27的有向图，那么D^c就如图中所示的有向图. 给出有向图D的例子，它是弱连通的，但不是单连通的，且使得：

144

(a)D^c是强连通的.

(b)D^c是单连通的但不是强连通的.

(c)D^c是弱连通的但不是单连通的.

35. 在有n个顶点的完全对称有向图中，如果两个循环被认为是相同的，如果其中一个循环是通过改变开始顶点从另一个循环得到的. 确定在这样的有向图中长度为k的不同循环的数量.

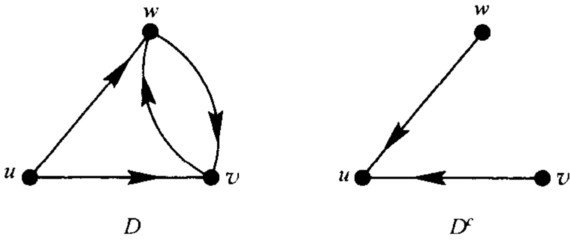


图3.27　有向图D和它的补有向图D^c

3.3　图着色及其应用

3.3.1　一些应用

在例1.4中，我们考虑了州立法委员会的会议调度问题. 并把这一问题翻译成关于图的问题. 本节，我们把这一图问题阐述为一个图着色问题. 我们讨论图着色的几个应用. 在下一节，我们将运用第2章的计数工具计数图着色的数量.

【例3.11 立法委员会的会议调度(再探例1.4、2.33、2.35)】 在会议调度问题中，我们给出了一个图G，其中G的顶点是需要安排会议时间的所有委员会，当且仅当两个委员会有公共成员时就用一条边连接这两个委员会. 现在，我们希望以这种方法给每一个委员会指定一个会议时间，使得如果两个委员会有公共成员，即如果对应顶点由G中的一条边连接，那么这些委员会得到不同的会议时间. 取代指定会议时间，考虑给G中的每一个顶点指定一种颜色(对应于一个会议时间)，使得如果两个顶点由一条边连接，那么它们就得到不同的颜色. 委员会调度是一个非常好的图着色(graph coloring)(也称*顶点着色*或*着色*)例子：着色一个图的顶点，使得邻接顶点得到不同的颜色. (图的边着色将在练习中讲述.)如果使用至多k种颜色就可以实现G的着色，那么我们称这样的着色是G的*k着色*且说G是*k可着色的*. 使得G是k可着色的最小数字k称为G的*色数*(chromatic number)，记为$\chi(G)$. 为了说明这些概念，返回图1.1中的图并称这个图为G. G的一个3着色如图3.28所示. 使用的三种颜色是红色、白色和蓝色. 注意，这个图也有一个4着色，实际上，根据定义，该图表示了一个4着色——我们不要求使用每一种颜色. 使用四种颜色的一个4着色是将图3.28中的财政顶点(实际上是任意顶点)改成绿色. 我们总是可以增加所用的颜色数目(直至顶点数目). 因此，问题的关键是寻找我们所使用的最小颜色数，即$\chi(G)$. 这里，$\chi(G)$显然等于3，因为三个顶点教育、住宅和健康必须得到不同的颜色. 本节描述图着色的应用. 我们在例1.4中提到，与调度委员会会议带有相同性质的其他应用包括大学中的期末考试调度和教室安排，工厂中的工作安排调度等问题，以及许多类似的问题. ■

145

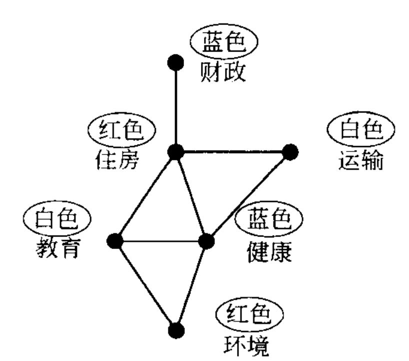


图3.28　图1.1中图G的一个3着色. 指定给顶点的颜色在圆圈中给出

【例3.12 变址寄存器和优化编译器(Tucker[1984])】 在一个优化编译器中，当程序在循环内计算时，把频繁使用的变量的值临时存储于中央处理器的变址寄存器中，而不是存储于正规的内存中，这样做是更加高效的. 我们希望知道，关于给定的循环，存储需要多少个变址寄存

器？我们设在这一循环内的变量是一个图 G 的顶点，而且，如果在循环内的某个步骤两个变量都必须存储的话，在 G 中的两个变量之间画一条边．如果两个变量可以由 G 中的一条边连接起来，那么我们希望以某种方式给每一个变量分配一个变址寄存器．两个变量必须指定给不同的寄存器．所需的寄存器的最小数量由 G 的色数给出． ■

146

【例 3.13 信道分配】 把一个信道分配给某区域的电视发射器供其使用．如果两个发射器相互在 100 英里的范围内，那么他们必须得到不同的信道．分配信道的问题可以看成是一个图着色问题．设一个图 G 的顶点是发射器，由一条边连接两个发射器，当且仅当他们相互在 100 英里的范围内时．分配一种颜色(信道)给每一个顶点使得两个顶点由一条边连接时，他们得到不同的颜色．对于给定的区域，需要多少个信道？这个数是 G 的色数．(关于图着色对电视频道、电台频率分配等更多的内容可以参见 Cozzens 和 Roberts[1982]、Hale[1980]、Opsut 和 Roberts[1981]、Roberts[1991]、van den Heuvel，Leese 和 Shepherd[1998]、Welsh 和 Whittle[1999].) ■

【例 3.14 安排垃圾卡车路线】 接下来考虑由纽约市卫生部提出的路线安排问题(参见 Beltrami 和 Bodin[1973]、Tucker[1973])㊀．需要清楚的是，这里讨论的技术可以应用于其他路线安排问题，例如，送奶路线和航空路线等．垃圾卡车可在给定的一天内访问若干场所．这样的一辆卡车的*游历*是它在给定的一天内所访问的场所的一个调度(次序)，且遵循这样的限制：这一游历可以在一个工作日完成．我们希望寻找带有下列性质的游历集合：

1. 每一个场所 i 在一周内被访问给定数 k_i 次．

2. 这一游历可以分摊到一周的六天内(星期天是假日)，使得(a)在一天内没有一个场所被访问两次㊁且(b)任意一天分配得到的游历数不超过卡车数．

3. 对于所有卡车来说，所参与的总时间是最小的．

为解决这一问题提出的一个方法，开始于任意给定的游历集合，并就总时间而言不断地改进这一集合(在现存的技术状态下，这一方法接近于一个最小集合，但不是总能够达到一个最小集合)．在每一步，给定的已改进游历集合必须得到测试，以便看它是否能以某种方式分割使得满足条件(2a)，即以某种方式在一周的六天内分摊，使得在一天内没有被访问两次的场所．因此，我们需要一个有效的"可分割性"测试，它可以反复地使用．这样一个测试的公式化可归结为一个图着色问题，而且这一问题将是我们所讨论的问题(对于整个问题的处理的描述，建议读者参看 Beltrami 和 Bodin[1973]及 Tucker[1973])．

为了测试一个给定的游历集合是否可以分割，使得满足条件(2a)，我们定义一个如下所示的*游历图*(tour graph)G．G 的顶点是这一集合中的游历，两个不同的游历用一条边连接起来，当且仅当它们服务于某个共同的场所．于是，这个给定的游历集合能够以某种方式在一周的六天内分摊，使得条件(2a)得到满足，当且仅当顶点 $V(G)$ 的集合能够分成六类，使其具有性质：G 中没有边连接同一类中的顶点．使用颜色描述这一问题会很方便．这一分割中的每一类被指定给六种颜色中的一种，我们需要一种顶点的颜色分配，使得一条边不连接同一颜色的两个顶点．㊂现在，关于游历问题可以改述如下：这一游历图是 6 可着色的吗？ ■

147

【例 3.15 舰队维修】 为了定期维修，我们按计划的时间让交通工具(汽车、卡车、船、飞机)进入维修厂．在这一维修厂中，在时间重叠期间，必须为两个交通工具指定不同的空间．这一维

㊀ 图论对卫生的其他应用将在 11.4.3 节中讨论．

㊁ 包括要求(a)是要确保及时清理垃圾以保证没有积存．

㊂ 这一思想源自 Tucker[1973]．

修厂需要多少空间呢？这一问题也可以阐述为图着色问题. 令图 G 的顶点是安排进行维修的交通工具，且两个顶点由一条边连接起来，如果它们的维修计划时间有重叠. 指定一种颜色(空间)给每一个顶点，使得有边连接的两个顶点得到不同的颜色(空间). 这一问题的答案就是 $\chi(G)$. ■

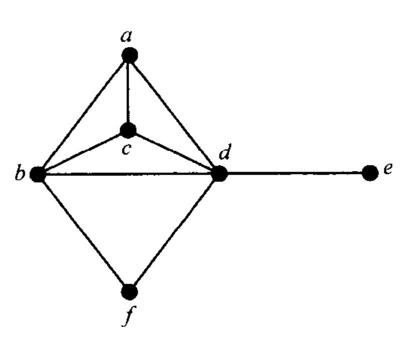


图 3.29 一个团数为 4 的图

【例 3.16 团数和色数】 假设 G 是一个图，G 中的一个团(clique)是一个顶点集合，其中每一个顶点都与团中其他顶点由一条边相连. 例如，在图 3.29 的图 G 中，$\{a, b, c\}$，$\{d, e\}$ 和 $\{a, b, c, d\}$ 就是团. G 的团数(clique number)，$\omega(G)$，是 G 的最大的团的大小. 在我们的例子中，$\omega(G)=4$. 因为在一个团内的所有顶点必须得到不同的颜色，这意味着 $\chi(G) \geqslant \omega(G)$. 你能给出一个例子，其中 $\chi(G) > \omega(G)$ 吗? ■

148

【例 3.17 色数和独立数】 假设 G 是一个图且 W 是 G 的顶点集合的一个子集. W 称为 G 的一个*独立集合*(independent set)，如果 W 中的任意顶点间都没有边. *独立数*(independence number)$\alpha(G)$ 是 G 的最大的独立集合的大小. 例如，在图 3.29 中，$\{a, f\}$ 和 $\{e, c, f\}$ 是独立集. 不存在四个顶点的独立集，所以 $\alpha(G)=3$. 我们假设 G 的顶点已被着色. 两个相同颜色的顶点之间不可能有边，所以给定颜色的所有顶点定义一个独立集. 因此，以 $\chi(G)$ 种颜色对 G 的 $n=|V(G)|$ 个顶点着色就把这些顶点分成 $k=\chi(G)$ 个“颜色类”，每一个颜色类定义一个独立集. 这样的独立集的平均大小是 $n/k=|V(G)|/\chi(G)$. 因此，根据鸽巢原理的应用(推论 2.15.2)，至少存在一个大小至少是 n/k 的独立集，即

$$\alpha(G) \geqslant \frac{|V(G)|}{\chi(G)}$$

或

$$\chi(G)\alpha(G) \geqslant |V(G)|. \tag{3.9}$$ ■

【例 3.18 课程调度】 假设一所大学让它的教授们在一周的任意时间内安排他们的课程. 假设每一门课程需要占据三个学时. 例如，一个学期的课程可以是如下形式：

1999 秋

数学 027：{星期一 3～4，星期四 1～2：15，星期五 3～3：45}

经济 321：{星期二 3～4：30，星期四 3～4：30}

统计及处理分析 114：{星期一 8～9，星期三 8～9，星期五 8～9}

⋮

学生如何挑选互不重叠的课程呢? 这所大学可以构建一个图，其中每一个顶点代表该大学开设的一门不同的课程. 如果两门课程在时间上有任何重叠，在相应两个顶点之间画一条边. 选 n 门课程的一个学生必须选择 n 个顶点，这些顶点形成这个图中的一个独立集合. ■

【例 3.19 地图着色】 对地图进行着色的问题是一个古老的重要问题，它已成为促进图论发展的重要诱因之一. 为了解释地图着色问题，让我们考虑图 3.30 中的地图. 理想的做法是以某种方式给地图上的国家着色，使得两个有公共边界的国家得到不同的颜色. 当然，可以为每一个国家着不同的颜色. 然而，很多年来，绘图师一直对使用尽可能少量的颜色着色地图感兴趣. 我们可以使用红色着色国家 1(参见图 3.31)来开始着色图 3.30 中地图上的国家. 然而，因为国家 2 与国家 1 共享边界，所以必须得到不同的颜色，例如蓝色. 国家 3 与目前已着色的国家中的每一

149

个都共享边界，所以它必须得到一种不同的颜色，例如绿色．国家 4 与前三个图家都共享边界，所以它也必须得到第四种颜色，例如黄色．国家 5 与国家 1，2 和 4 共享边界，但与国家 3 不共享边界．因此，可以使用绿色着色国家 5．最后，国家 6 不能是蓝色或绿色．在图 3.31 中，我们给它着上了红色．注意这张地图使用了 4 种颜色着色．没有人曾发现需要用多于四种颜色着色的地图，条件是“地图”和“边界”被精确定义，不存在诸如有分成两块的国家，共同边界是一个点的国家等．100 多年来，人们猜想每一张地图都可以用四种或少于四种颜色着色．然而，尽管世界上一些最优秀的数学头脑付出了努力，但是这一**四色猜想**(four-color conjecture)一直没有得到证明或推翻，四色问题一直没有得到解决．最终，于 1977 年，这一四色猜想得到了证明(参见 Appel 和 Haken[1977]、Appel，Haken 和 Koch[1977])．四色定理的原始证明包括使用高速计算机来检查某些特殊情况，而且还包括大约 1200 小时的计算机时间(最近的工作已导致了四色定理的“更简单”的证明；可参见 Robertson 等[1977])．

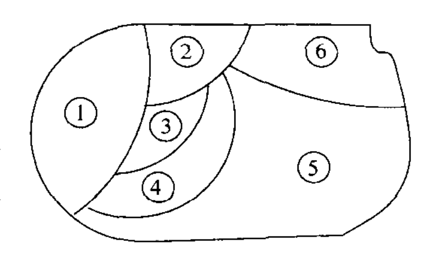


图 3.30　一张地图

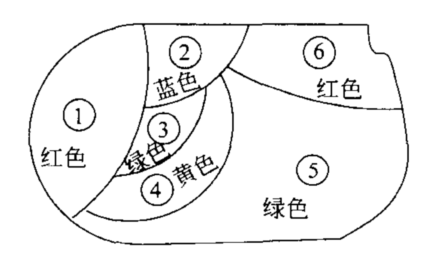


图 3.31　图 3.30 中地图的一种着色

在处理地图着色问题和地图的 k 着色时，关键步骤之一是把地图着色问题翻译成等价的但在某种程度上更易处理的问题．设每一个国家的首都由点表示．如果两个国家共享边界，那么用一条直线(虚线)连结其相应的两个首都．这就给出图 3.32的线．在图 3.33 中，我们只使用首都以及连接这些首都的线重新绘制了这一地图．它定义了一个图．不是着色整个国家，而是考虑只着色它的首都．以图 3.33 中的图为例，要求是如果两个首都，或顶点由一条边连接，那么它们必须得到不同的颜色．因此，一个地图是用 k 种颜色可着色的，当且仅当相对应的图是 k 可着色的．■

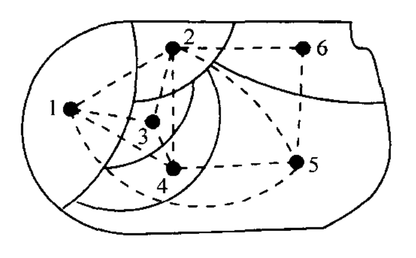


图 3.32　虚线连接两个首都，当且仅当它们所对应的国家共享一个边界

图着色及其扩展除在这里所描述的，在其他方面也有很多应用，例如，在计算机科学中的时间共享，在运输学中的定向信号灯和运筹学中的各种调度及维修问题．对于这些问题的一些陈述可以参见 Opsut 和 Roberts[1981]及 Roberts[1991]．

150

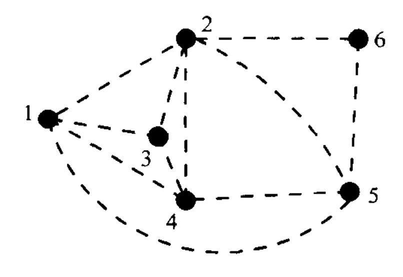


图 3.33　根据图 3.32 中的地图连接首都的图

3.3.2　平面图

图 3.33 中的图有这样的性质：除在这一图的顶点之外，没有相交的边．具有这一性质的图，或者有符合这一性质的等价(同构)重画图称为**平面**(planar)的．每一个地图给出一个平面图，且反过来，每一个平面图来自于一个地图．因此，四色定理可以描述为如下图论中的定理：每个平面图是 4 可着色的．图 3.34 中的第一个图是平面的，即使图上有相交的边．图 3.34 中的第二个图与第一个图等价(同构)且没有相交的边．图 3.34 中的第一个图是完全图 K_4．因此，K_4 是平面的．图 3.35a 的图 K_5 不是平面的．无论你如何定位这一平面中的五个点，都不可能使用直线把它们连接起来，且其中的任意两条直线都不相交．鼓励读者尝试一下．图 3.35b 的图是另一个非平面图的例子．这个图称为水光气图(water-light-gas

graph)，记为 $K_{3,3}$. 我们考虑三个房子和三种公用设施并尝试着把每一个房子与每一种公用设施连接. 没有不相交的直线是不可能实现这一点的. 再一次鼓励读者尝试一下. 确定一个图是否是平面图的问题有着广泛的应用. 例如，在电机工程中，平面图正好对应于可能的印刷电路. 在11.6.4节中，我们给出平面图在设备设计问题中的使用. Kuratowski[1930]证明在某种意义下，K_5 和 $K_{3,3}$是仅有的非平面图⊖.

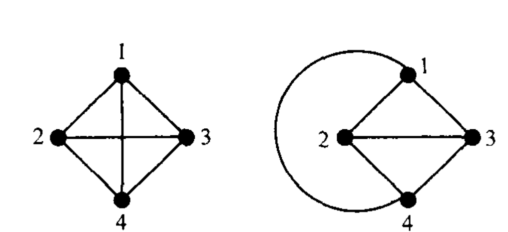


图 3.34 第一个图是平面的，因为可以通过第二个图来说明，而且由顶点标签所示，它们是同构的

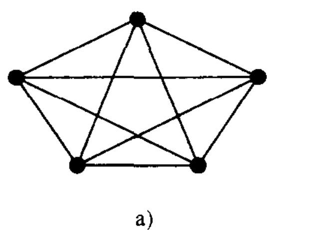
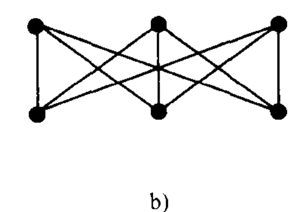



图 3.35 两个非平面图 K_5 和 $K_{3,3}$

为了明确在什么意义下 K_5 和 $K_{3,3}$是仅有的非平面图，我们说图 G'是从图 G 通过**细分**(subdivision)得到的，如果我们通过在 G 的一条边上加入顶点而得到 G'. 在图 3.36 中，图 G_i'都是从图 G_i 通过细分得到的. 两个图 G 和 G'被称为**同胚的**(homeomorphic)，如果二者可以从同一个图 H 通过一系列细分而得到. 例如，任意两个简单链是同胚的. 图 3.37 给出两个图 G 和 G'，二者都是从图 H 通过一系列细分得到的. 因此，G 和 G'是同胚的(同时，它们还同胚于 H).

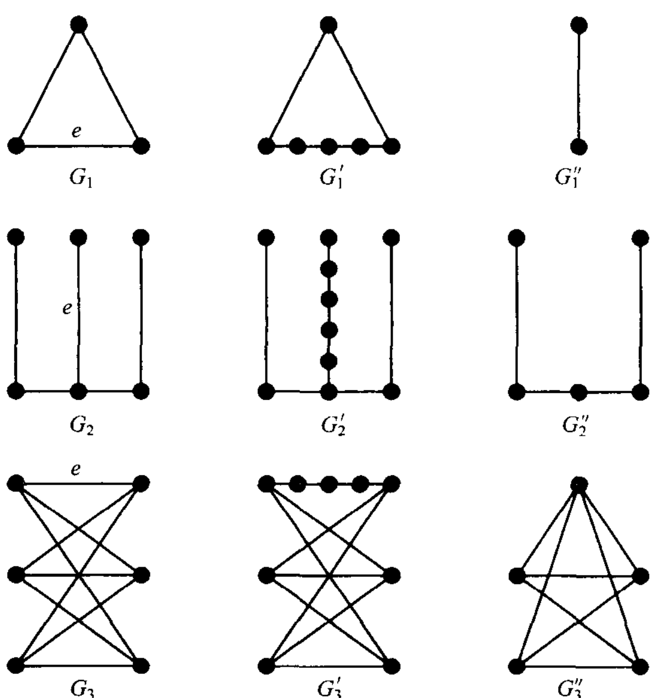


图 3.36 图 G_i'是通过边 e 的细分从图 G_i 得到的. 图 G_i''是通过收缩边 e 从图 G_i 得到的

⊖ 本小节的其余部分可以跳过.

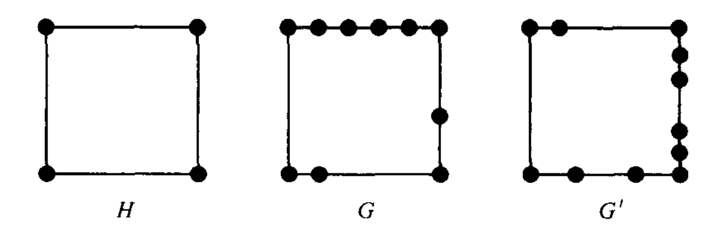


图 3.37 G 和 G' 是同胚的，因为它们每一个都是通过细分从 H 得到的

定理 3.2(Kuratowski[1930]) 一个图是平面的当且仅当它没有同胚于 K_5 和 $K_{3,3}$ 的子图㊀.

对于定理 3.2 的证明，我们建议读者参阅 Harary[1969]、Bondy 和 Murty[1976]或 Makarychev[1997]. 根据 Kuratowski 定理，图 3.38 的图 G 不是平面的，因为它同胚于 K_5，而且，图 G' 也不是平面的，因为它有一个子图 H 同胚于 $K_{3,3}$.

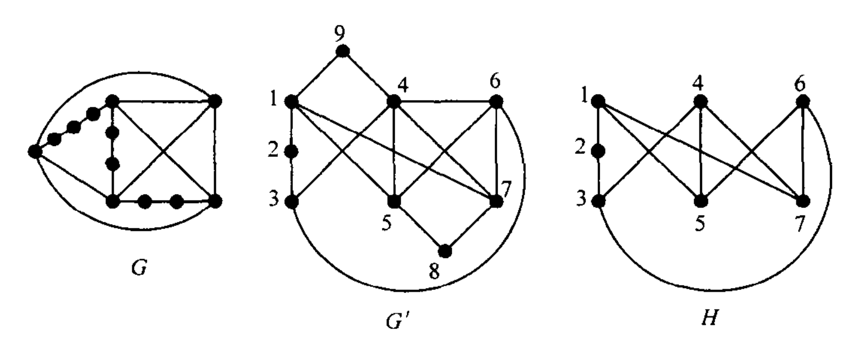


图 3.38 G 与 K_5 同胚，G' 有一个同胚且收缩到 $K_{3,3}$ 的子图 H

假设 $e=\{x, y\}$ 是 G 的一条边. 收缩(contracting)e 意味着等同两个顶点 x 和 y. 这一新的组合顶点与所有曾与 x 或 y 连接过的那些顶点连接. 如果 x 和 y 都曾与顶点 z 连接，那么只取一条从组合顶点到 z 的边. 在图 3.36 中，图 G_i'' 总是可以通过收缩 e 从图 G_i 得到. 为了得到 G'，把 G 收缩到 G' 意味着收缩 G 的一系列边. 使用收缩的平面图的另一个特征可以由下面的定理给出，这一定理源自 Halin[1964]、Harary 和 Tutte[1965]及 Wagner[1937].

定理 3.3 一个图 G 是平面的，当且仅当 G 没有可以收缩到 K_5 或 $K_{3,3}$ 的子图.

对于定理 3.3 的证明，可以参见 Harary[1969]，Bondy 和 Murty[1976]和 Makarychev[1997]. 根据定理 3.3，图 3.38 的图 G' 不是平面的，因为它有一个子图 H 可以通过收缩边$\{1, 2\}$或$\{2, 3\}$收缩到 $K_{3,3}$.

在结束本节之前，我们注意到 Kuratowski 定理并没有给出检测图的平面性的好算法. 然而，存在这样的算法. 对这样的算法的讨论，可以参见 Even[1979]. 特别地，对于二次[$O(n^2)$]算法可以参见 Demourcron，Malgrance 和 Pertuiset[1964]和 Klotz[1989]，而对于线性[$O(n)$]算法可以参见 Hopcroft 和 Tarjan[1974]、Booth 和 Lueker[1976].

3.3.3 计算色数

我们来研究一下各种图的可着色性. 图 K_4 显然可以使用四种颜色着色，但不能使用三种或更少的颜色着色. 因此，$\chi(K_4)=4$. 图 K_5 可以使用五种颜色着色，但不能以四种颜色或更少的颜色着色(为什么). 因此，$\chi(K_5)=5$. (这不是四色定理的反例，因为我们已指出 K_5 不是平面图，所以不会在地图中出现.)图 3.35 中的 $K_{3,3}$ 可用两种颜色着色：上面三个顶点着色为红色，

㊀ 没有必要是生成子图.

下面三个顶点着色为蓝色. 因为显然需要两种颜色，所以 $K_{3,3}$ 的色数是 2.

我们简单地回顾例 3.14 中的游历图问题. 一般地，为了运用这一过程寻找游历的最小集合，我们必须有一个算法，这一算法可以快速地重复使用，目的在于确定给定的图是否是 k 可着色的. 可是，没有这样的总是“好”的解决这一问题的多项式算法. 实际上，一般地，不知道是否存在一个“好”算法(在 2.4 节的意义下)来确定给定的图是否是 k 可着色的. 在 2.18 节的意义下，这一问题是 NP 完全的，所以在精确的意义下是一个很困难的问题. 因此，垃圾卡车路线问题可以简化成一个困难的数学问题. 然而，用精确的数学公式来描述可以使我们清楚为什么这一问题是一个难题，公式化也可以为我们提供解决这一问题的很多工具，至少在特殊情况下是这样. 需要注意的是，在现实世界中，只说一个问题是不可解的或是很难的是不够的. 想象一位拿着 500 000 美元薪水的顾问走进市长办公室，汇报说经过仔细的研究后，他得出结论说垃圾卡车路线问题是一个难题时的情景! 垃圾卡车必须定好路线. 所以在这样的情况下你能做什么呢? 答案是，你要开发部分解决方案，你要开发仅对某些特殊情况可应用的解决方案，你修改这一问题，或在某些情况下，你甚至可以“说谎”. 你可以使用未必是真的但似乎可以工作的结果说谎. 其中一个这样的结果就是**强完美图猜想**或**强伯格猜想**，这可追溯到 Claude Berge[1961, 1962](理解下面的推理并不需要知道这一猜想的内容，对于这一猜想的详细讨论可以参见 Golumbic[1980]，对于这一猜想的讨论以及它对垃圾卡车及路线的应用可以参见 Roberts[1976, 1978]或 Tucker[1973]). 正如 Tucker[1973]所指出的那样，如果这一猜想是真的，那么存在高效的算法，利用这个算法可以确定给定的图是否可以用给定数量的颜色着色. 至少在游历图问题中是这样的，在这一问题中，游历图只做局部变化而不是全局变化. 因此，Tucker 认为值得“说谎”并在确定垃圾卡车路线中值得使用强伯格猜想. 那么，会出现什么错误呢? Tucker 说，可能发生的最坏情况如下所示. 你运用这一猜想确定垃圾卡车路线并寻找一条对一周的六天可分配的路线，但是事实上，这一路线是不可分配的. 这种最坏的情况对数学是极大的好处: 我们发现了强伯格猜想的反例! 这一非凡的论证不再是必要的. 经过 40 年后，Chudnovsky 等[2000]证明了强伯格猜想(参见 Mackenzie[2002]).

151 ~ 154

3.3.4 2 可着色图

接下来让我们注意存在一个 k 的值，对于该值很容易确定 G 是否是 k 可着色的. 这一情况是 $k=2$[⊖]. 一个图是 2 可着色的当且仅当其顶点可以分成两类使得这个图中的所有边都连接两种不同类中的顶点. (为什么?)具有这种划分的图被称为是**二部的**(bipartite). 11.1 节描述的深度优先搜索过程给出了测试一个图是否是二部图的多项式算法(参见 Reingold, Nievergelt 和 Deo[1977, pp. 399-400]).

接下来我们描述 2 可着色图的一个非常有用的特征. 设 Z_p 是由 p 个顶点的单一回路组成的图. 图 3.39 给出 Z_3, Z_4, Z_5 和 Z_6. 容易证明 Z_4 和 Z_6 是 2 可着色的. 图 3.39 给出了二者的 2 着色. 显然，Z_3 和 Z_5 不是 2 可着色的. 这需要

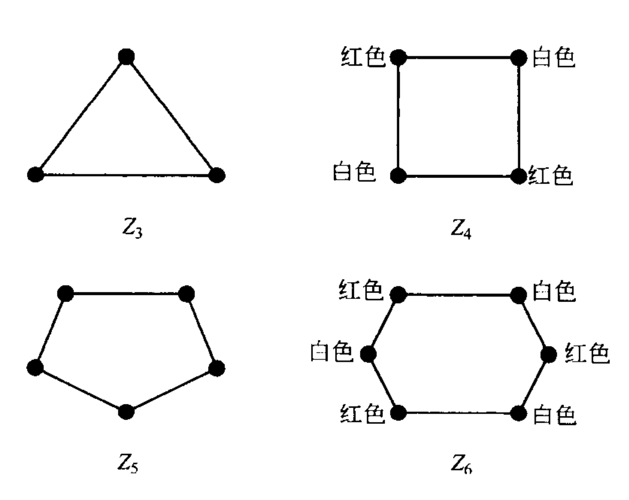


图 3.39 $p=3, 4, 5, 6$ 的图 Z_p

⊖ 对于存在确定 G 是否 k 可着色的好算法的其他情况的讨论可以参见 Garey 和 Johnson[1979]、Golumbic[1980]、Jensen 和 Toft[1995].

[155] 一点证明，我们把证明留给读者．一般地，很容易看到，Z_p 是 2 可着色的当且仅当 p 是偶数．

现在，假设我们从 Z_5 开始，尽可能增加一些边，然后增加一些新的顶点和一些连接这些顶点或把这些顶点和 Z_5 的顶点连接起来的边．我们也许得到诸如图 3.40 中的那些图．现在，其中的每一个图都不是 2 可着色的．因为整个图的 2 着色会自动给出 Z_5 的一个 2 着色．这是一个一般规则：任意图 G 的 k 着色是 G 的所有子图的 k 着色．因此，以 Z_5 为子集的任意图都不是 2 可着色的．对于 Z_3，Z_7 和 Z_9 也同样成立．如果 G 有奇数长度的任意回路，那么这个回路定义一个形如 Z_P 的子图，而且 p 是奇数；因此，G 不可能是 2 可着色的．这一陈述的逆也为真，我们将这一结果阐述为如下定理．

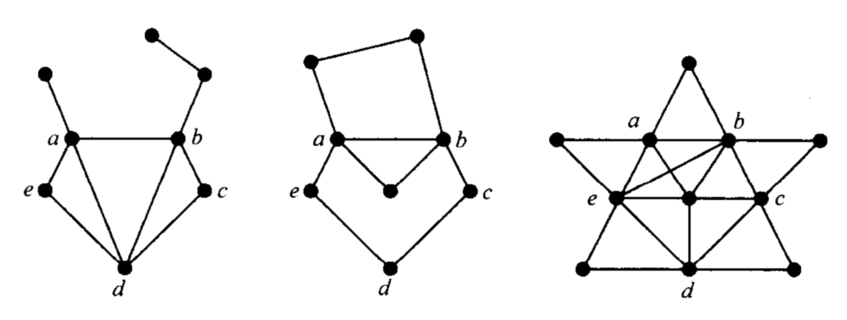


图 3.40　以 Z_5 为子图的图．Z_5 的顶点被标记为 a，b，c，d 和 e

定理 3.4(König[1936])　一个图是 2 可着色的当且仅当它没有长度为奇数的回路．

为了对定理 3.4 中的逆进行证明，我们从没有奇数长度回路的图 G 开始，给出寻找 G 的 2 着色的算法．我们可以假设 G 是连通的(否则，我们可以分别着色每一个连通分支)．任意挑选一个顶点 x．把它着色为蓝色．把 x 的所有邻居着色成红色．对于其中的每一个邻居，着色它的没有着色的邻居为蓝色．继续这样做下去直到所有顶点都被着色．这一算法可以通过图 3.41 来说明，图 3.41 是连通的且没有奇数长度的回路．这里，选择的顶点 x 是 a，2 着色如图所示．

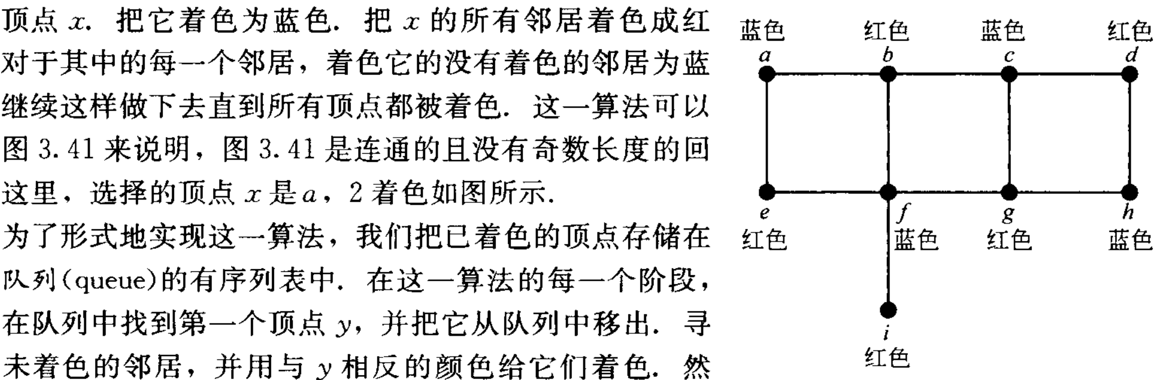


图 3.41　没有奇数长度回路的连通图．首先着色顶点 a 得到一个 2 着色

为了形式地实现这一算法，我们把已着色的顶点存储在称为队列(queue)的有序列表中．在这一算法的每一个阶段，我们在队列中找到第一个顶点 y，并把它从队列中移出．寻找 y 未着色的邻居，并用与 y 相反的颜色给它们着色．然后，我们把这些邻居加到队列的尾端．我们继续这一过程直到所有顶点都被着色．这一算法被形式地描述为算法 3.1．这里，Q 是所述的队列．[156~157]

算法 3.1：2 着色

输入：没有奇数长度回路的连通图 $G=(V, E)$．

输出：使用两种颜色红和蓝对 G 的顶点的着色．

步骤 1. 初始化，V 的所有顶点都没有着色且 Q 是空的．

步骤 2. 在 V 中挑选顶点 x，着色 x 为蓝色，并把 x 放入到 Q 中．

步骤 3. 令 y 是 Q 中的第一个顶点．将 y 从 Q 中移出．

步骤 4. 寻找 y 的所有未着色的邻居．使用与 y 的颜色相反的颜色着色每一个邻居．以任意顺序把它们加到 Q 的尾部．

步骤 5. 如果所有顶点都被着色，停止．否则返回到步骤 3．

为了说明算法 3.1，考虑图 3.41 中的图. 这些步骤概括在表 3.2 中. 挑选出的顶点 x 是 a，着色 a 为蓝色，且把 a 放入到队列中. 寻找 a 的未着色的邻居，即 b 和 e. 把它们着色成红色(这与 a 的颜色相反). 把 a 从队列 Q 中移出，并以 b，e 的顺序把 b 和 e 加入到 Q 中. 挑选出 Q 中的第一个顶点；这里是 b. 把它从 Q 中移出. 寻找它的未着色邻居；它们是 c 和 f. 使用与 b 的颜色相反的颜色着色它们，即蓝色. 以任意顺序把它们加到 Q 的尾端. 如果 c 是首先加入的，那么 Q 现在是 e，c，f. 把 Q 中的第一个顶点移出，这里是 e. 它没有未着色邻居. Q 现在是 c，f. 继续挑选 Q 中的第一个顶点，即 c，把它移出. 寻找它的未着色邻居，这里是 d 和 g，并使用与 c 相反的颜色着色它们，这里是红色. 把 d 和 g 加到 Q 的尾端，例如 d 是第一个加入的. Q 现在是 f，d，g. 接下来把 f 从 Q 的前端移出，把它的未着色邻居 i 着色为红色，并把 i 加到 Q 的尾端. Q 现在是 d，g，i. 最后，把 d 从 Q 的前端移出，着色它的未着色邻居 h 为蓝色，并把 h 加到 Q 的尾端. 停止，因为到此所有顶点都已着色.

表 3.2 把算法 3.1 运用到图 3.41 中的图

当前考虑的顶点	被着色顶点	新队列 Q	当前考虑的顶点	被着色顶点	新队列 Q
	a(蓝色)	a			
a	b，e(红色)	b，e	c	d，g(红色)	f，d，g
b	c，f(蓝色)	e，c，f	f	i(红色)	d，g，i
e	无	c，f	d	h(蓝色)	g，i，h

我们使用的访问所有顶点的过程被称为*广度优先搜索*(breadth-first search). 这是一个非常高效的计算机过程，它在图论中有很多应用. 我们将在 11.1 节讨论检测图的连通性的算法时，讨论广度优先搜索及称为深度优先搜索的相关过程. 在 2.4 节和 2.18 节的意义下，算法 3.1 是“好”算法. 不难证明它的复杂度的阶是 $n+e$，其中 n 是这个图的顶点数量，而 e 是其边数. 因为一个图至多有 $\binom{n}{2}$ 条边，所以我们如 3.2.3 那样推理得出结论：

158

$$e \leqslant \binom{n}{2} = \frac{n(n-1)}{2} \leqslant n^2.$$

因此，算法 3.1 至多需要步骤数为 $n+n^2$ 的阶，它是 n 的多项式(使用 2.18 节的记法，这个算法是 $O(n^2)$).

为了说明算法 3.1 的功效，我们必须证明每一个顶点最终都被着色，并且证明用这种方法可得到一个图着色. 图 G 的连通性确保每一个顶点最终被着色(我们省略这一事实的形式证明). 为了证明可以得到一个图着色，假设 u 和 v 在 G 中是邻居. 它们能得到相同的颜色吗？检查它们不能得到相同颜色的最简单方法是，在一个连通图内定义顶点 a 和 b 之间的距离 $d(a, b)$ 是它们之间的最短链的长度. 于是，能够证明如果 $d(x, z)$ 是奇数则顶点 z 着色成红色，如果 $d(x, z)$ 是偶数则顶点 z 着色成蓝色⊖(这一证明留作练习 31). 现在，如果两个邻居 u 和 v 都被着色成红色，那么存在一条从 x 到 u 的最短链 C_1，其长度为奇数，及一条从 x 到 v 长度为奇数的最短链 C_2. 因此 C_1 加上边 $\{u, v\}$ 加上(后向的) C_2 形成一个从 x 到 x 的长度为奇数的闭链. 但是，如果 G 有一条奇数长度的闭链，它必定有一个奇数长度的回路(练习 32). 因此，我们得到一个矛盾. 如果 u 和 v 都被着色成蓝色，我们也可得到类似的矛盾.

记住，算法 3.1 和定理 3.4 只适用于 2 着色. 一般的图着色问题是 NP 完全问题；对于任意固定的 $k \geqslant 3$，不存在确定 k 可着色性的多项式算法，更不用说寻找实用的最优着色.

⊖ 定理 3.4 的最短证明是简单地用这种方法定义着色.

3.3.5 图着色变形

存在很多图(顶点)着色的变形. 下面的三个例子只是揭开图论中这一急速发展领域的表皮.

【例 3.20 信道分配和 T 着色问题】 回想一下例 3.13 的信道分配问题. 与这一问题相关的图把发射器当作顶点，且两个顶点之间有一条边，如果所对应的发射器与另一个发射器间的距离在 100 英里以内. 使用尽可能少的颜色着色这一图等价于解决信道的分配问题.

159

然而，在某种情况下，不仅"挨着的"发射器不能得到相同的信道，而且他们不能得到相差某些特定值的信道. 我们称这一"禁止"的差集合为 T-集合(T-set)，记为 T. 例如，对于 UHF 电视，相距不足 55 英里的发射器不能得到其差为 T 集合中任意值的信道，其中 $T=\{0, 1, 7, 14, 15\}$. 给定一个 T 集合和一个图 G，G 的顶点着色称为 G 的 T 着色，如果在这一着色中，邻接顶点没有绝对差在 T 中的颜色(绝对差是指 $|c_i-c_j|$，其中 c_i，c_j 是颜色). 图 3.42 给出了带有 T 集合 $T=\{0, 3, 4, 7\}$的一个图的 T 着色例子. 注意，邻接顶点的颜色绝对差等于 1，2，8，9，10，其任意一个都不在 T 集合内. 我们假设所有 T 集合都包含 0. 否则，所有顶点都被着色相同颜色的毫无意义的情况会组成一个 T 着色. ■

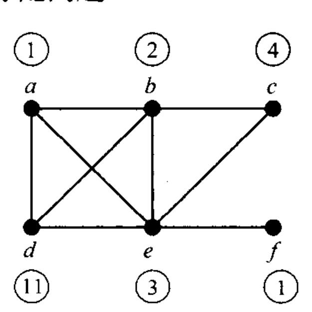


图 3.42　带有 T 集合 $T=\{0, 3, 4, 7\}$的一个图的 T 着色

在例 3.20 中，我们可以用颜色 1，100，500，1000 和 2000(取代 1，2，3，4 和 11)容易地着色图 3.42 中的顶点，产生一个 T 着色. 然而，这不是一个高效 T 着色. 那么，一个高效 T 着色意味着什么呢? 正如最小化颜色数量被用作高效图着色的标准一样，我们需要高效 T 着色的标准. 关于信道分配，有时候我们所关心的是所用的信道总数，而有时候我们又关心的是所用的信道范围. 因此，有时候，我们对于 T 着色中所用的颜色总数量感兴趣，而有时候，又对所用的颜色范围更感受兴趣.

一个 T 着色的阶(order)指的是所用的颜色总数量. 图 3.42 中的图的 T 着色阶等于 5，因为使用了五种不同的颜色：1，2，3，4，11. T 着色图 G 所需要的最小颜色数量(即最小阶)称为 G 的 T 色数，记为 $\chi_T(G)$. 图 3.42 所使用的 T 着色在这一意义下不是最有效的，因为可以使用 4 种颜色着色(但不是 3). 顶点 a，b，d 和 e 都需要不同的颜色，因为任意两个顶点可以由一条边连接起来. 因此，$\chi_T(G)\geqslant 4$. 顶点 c 和 f 都可以与 a(或 d)有相同的颜色，产生一个阶为 4 的 T 着色. 所以，如果 G 是图 3.42 的图，且 $T=\{0, 3, 4, 7\}$，那么 $\chi_T(G)=4$.

160

如下面的定理所示，图 G 的 T 着色的最小阶 $\chi_T(G)$并非一个新参数.

定理 3.5(Cozzens 和 Roberts[1982])　*对于所有图 G 和任意 T 集合 T，有*

$$\chi_T(G)=\chi(G).$$

证明　因为我们假设 0 包含在每一个 T 集合中，G 的任意 T 着色将(起码)是一个图着色. 因此 $\chi_T(G)\geqslant\chi(G)$.

接下来，可以使用下面的方法把 j 种颜色的 G 的任意图着色转化成 j 种颜色的 T 着色. 不失一般性，我们假设图着色中的 j 种颜色是 1，2，…，j. 使用颜色 $i\cdot(t+1)$取代颜色 i，其中 t 是 T 集合 T 的最大元素. 这些新颜色形成 G 的一个 T 着色(参见练习 37). 因此，使用 $\chi(G)$种颜色的图着色可以转化成使用相同数量颜色的 T 着色. 因此，$\chi_T(G)\leqslant\chi(G)$.

我们证明了 $\chi_T(G)\geqslant\chi(G)$和 $\chi_T(G)\leqslant\chi(G)$. 因此有

$$\chi_T(G)=\chi(G).$$

■

高效性的一个更普通的标准必须与 T 着色中所使用的颜色范围有关. 在信道分配中，很多时候，重要的不是需要分配多少信道，而是信道的总需求是否能适合分配的带宽. 我们定义 T 着色的*跨度*(span)是所使用颜色的最大值与最小值的差. 图 3.42 中的 T 着色的跨度等于 10，它得自于 11-1，11 是这一 T 着色中所使用的颜色的最大值，而 1 是最小值. 图 G 的所有 T 着色的最小跨度称为 G 的 *T 跨度*(T-span)，记为 $sp_T(G)$. 图 3.42 所使用的 T 着色关于跨度是最高效的. 假设 1 是任意 T 着色使用的最小颜色，把颜色只局限于 1，2，…，10 将不能为这个图构成一个 T 着色(练习 38).

图的 T 色数和 T 跨度是一个不容易确定的值. 使用 2.18 节的记法，其每一个的计算都属于 NP 完全问题类. 然而，对于某些图和特殊的 T 集合，已找到 T 跨度的精确值(例如，参见 Bonias[1991]、Liu[1991]、Raychaudhuri[1985]、Tesman[1993]和 Wang[1985]).

【例 3.21 任务分配和集合着色问题】㊀ 诸如构建一架飞机等大型且复杂的任务实际都是由很多子任务组成的. 其中一些子任务是不相容的，因此不能在同一时间内进行. 例如，一些子任务可能需要相同的工具、资源、飞机棚空间等等. 任务分配问题就是调度子任务使得只有相容的子任务才能安排在重叠时间内.

为了用图理论阐明这一问题，我们令每一个顶点代表这个大型任务的一个子任务. 如果两个子任务不相容，则在它们之间连一条边. 于是，因为每一个子任务都需要一个“时间段”，我们不是给每一个顶点只分配一种颜色，而是分配一组颜色，这组颜色代表这一子任务所需要的时间. 如果 x 是一个顶点，我们记分配给 x 的集合为 $S(x)$. 如前面一样，我们将要求邻接顶点不能有相同的颜色. 据此，我们的意思是如果 x 和 y 邻接，那么 $S(x)$和 $S(y)$必须没有任何公共成员，即 161

$$S(x) \cap S(y) = \varnothing.$$

这样的一个着色称为这个图的*集合着色*(set coloring). 图 3.43 给出了图 Z_5 的集合着色例子. 注意这些集合不必有相同的大小；在这一例子中它们就没有相同的大小. 当对顶点着色时，最小化所使用的颜色总数量是集合着色中高效性的标准. ■

为了给集合着色更多结构，我们考虑所有已分配的集合大小相同的特殊情况(Gilbert[1972]在关于涉及移动式无线电通信的频率分配的问题中引入这一思想). 一个 *k 元组着色*(k-tuple coloring)指的是一个集合着色，其中每一个顶点被分配 k 个颜色的集合(如果每一个集合只包含一个元素，即如果我们有一个 1 元组着色，那么这个集合着色就是普通的图着色). 图 3.44 给出对图 Z_5 使用 8 种颜色的 3 元组着色的例子.

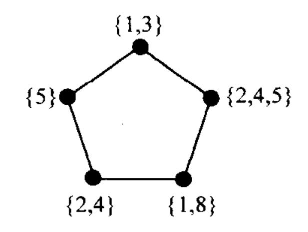


图 3.43 Z_5 的集合着色

k 元组着色一个图需要多少种颜色？如果这个图有 n 个顶点，一个大致的上界是 kn 种颜色(给每个顶点一个完全不同的 k 种颜色集合). 然而我们可以做得更好. 下面的定理给出了基于图的色数而非图的顶点集合大小的上界. 定理 3.6 的证明留作练习(参见练习 59).

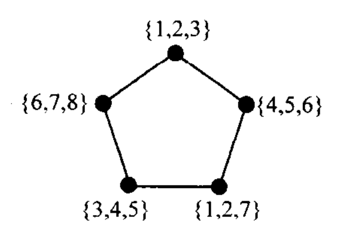


图 3.44 Z_5 的 3 元组着色

定理 3.6 *图 G 至多使用 $k \cdot \chi(G)$ 种颜色进行 k 元组着色.*

k 元组着色一个图所需要的最小颜色数量被称为 G 的 k 元组

㊀ 此内容来自于 Opsut 和 Roberts[1981].

色数(k-tuple chromatic number)，记为$\chi_k(G)$. 根据定理3.6，我们知道$\chi_k(G)\leqslant k\cdot\chi(G)$. 事实上，对于很多不同类的图，$\chi_k(G)$的上界是$\chi_k(G)$的精确值，但是一般情况下这不一定成立. 我们已证明可以使用8种颜色3元组着色Z_5，这比$3\cdot\chi(Z_5)=3\cdot3=9$好(Irving[1983]已证明寻找$\chi_k(G)$是一个NP完全问题). 事实上，8是图$Z_5$的3元组色数. 设$a$和$b$是任意一对邻接顶点. 那么，$a$和$b$不能有任何共同颜色. 不失一般性，使用颜色1，2，3着色顶点a，使用颜色4，5，6着色顶点b. 于是很容易证明只使用7种颜色无法给余下顶点进行3元组着色.

162

使用定理3.6中的上界可正确确定k元组色数的一类图是完全图. 因为在完全图内，任意两个顶点是邻接的，k元组着色中的颜色无法重复. 因此，我们有下面的定理.

定理3.7　*对于任意n和k，$\chi_k(K_n)=k\cdot\chi(K_n)=kn$.*

对于k元组的其他参考，可以参见Brigham和Dutton[1982]、Geller[1976]、Roberts[1991]和Scott[1975].

【例3.22 调度立法委员会会议(再探例1.4)：列表着色】　在例1.4中，我们考虑了一个调度问题，其中每一位委员会主席都提供一份可以接受的会议时间列表. 没有这一额外限制时，这一调度问题可以翻译成例3.11的图着色问题. 带有这一限制时，这一问题就变成寻找一个图着色的问题，其中分配给一个顶点的颜色是从可能接受的颜色列表中选出来的. 在很多其他应用中也出现相同种类的问题，例如，在信道分配中，给定发射机的用户可能指定一个可接收信道的列表. 设$L(x)$表示分配给顶点x的颜色列表. L称为G的*列表分配*(list assignment). 对G的一个顶点着色，使得分配给顶点x的颜色来自于列表$L(x)$，那么这个顶点着色称为L*列表着色*(L-list coloring).

图3.45a给出了有列表分配的图$K_{3,3}$的例子. 对于这一分配你能寻找到一个列表着色吗？顶点a必须被着色为1或2. 如果它被着色1，那么顶点d必须被着色为3，因为顶点a和顶点d之间存在一条边. 类似地，顶点f必须被着色为2. 但是没有给顶点c留下颜色. 如果顶点a被着色2，也存在类似的问题. 因此，我们证明了对于这一列表分配，这个图不是列表可着色的. 然而，如果分配给顶点的列表如果图3.45b所示，那么如表3.3所示的着色颜色给出一个具有列表着色功能的选择.

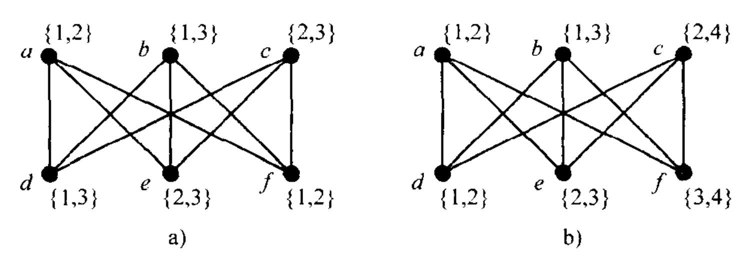


图3.45　$K_{3,3}$的两个列表分配

表3.3　图3.45b的列表分配和图的列表着色

顶点	a	b	c	d	e	f
列表	{1，2}	{1，3}	{2，4}	{1，2}	{2，3}	{3，4}
选择	1	1	4	2	3	3

■

【例3.23 修正列表的列表着色】　在委员会调度的例子中(例3.22)，如果不存在列表着色，你将做什么呢？我们可以要求某些人接受不在他们原来列表中的一些颜色. 考虑这一问题的一

163

个简单方法是允许某些人 x 扩展他们的列表 $L(x)$，向其中(从可能的颜色中)加入额外的颜色. 必须这样做的最小人数是多少? 我们说带有列表分配 L 的 G 是 p **可加的**(p-addable)，如果我们能够确定 G 中的 p 个不同顶点 x_1，x_2，…，x_p 和(不必不相同)$\bigcup L(x)$中的颜色 c_1，c_2，…，c_p，使得如果对于 $i=1$，2，…，p 有 $L'(x_i)=L(x_i)\cup\{c_i\}$，而对于其他顶点 x 有 $L'(x)=L(x)$，那么存在带有列表分配 L'的 G 的列表着色. 我们对 $I(G, L)$的计算感兴趣，即带有列表分配 L 的 G 是 p 可加的最小 p. 作为一个简单例子，考虑图 3.46 所示的图 G 和列表分配 L. 于是，不存在列表着色，因为顶点 b 和 d 在任意的列表着色中都必须得到不同的颜色. 然而，把颜色 2 加入到 $L(d)$中使得这成为列表可着色. 因此，带有 L 的 G 是 1 可加的且 $I(G, L)=1$.

图 3.46 一个 1 可加的图 G

考虑图 $G=K_{10,10}$，它的顶点有两类 A 和 B，每一类都有 10 个顶点，且 A 中的每一个顶点 x 与 B 中的每一个顶点 y 之间有边. 在 A 的顶点上，使用集合{1，2，3，4，5}的 10 个 2 子集作为集合 $L(x)$，对于 B 的顶点也是如此. 我们将证明 $I(K_{10,10}, L)=4$. 假设我们把一种颜色加入某些集合$L(x)$得到 $L'(x)$，使得存在 L'的 $K_{10,10}$的列表着色. 假设这个列表着色在 A 上使用 r 种颜色，在 B 上使用 s 种颜色. 那么显然有 $r+s\leqslant 5$. 现在，A 上的$\binom{5-r}{2}$个集合不使用这 r 种颜色，所以至少 A 上的$\binom{5-r}{2}$个集合需要加入一种颜色. 类似地，至少 B 上的$\binom{5-s}{2}$个集合需要加入一种颜色. 因为 A 上的所有其他集合都有 r 种颜色中的一种颜色，而且对 B 也同样如此. 所以加入的颜色也形成一个着色. 因此下式成立：

164

$$I(K_{10,10},L)\leqslant\binom{5-r}{2}+\binom{5-s}{2},$$

对于某个 r 和 s，上式为等式. 事实上，$r=3$ 和 $s=2$ 时可得到等式. 因此

$$I(K_{10,10},L)=\binom{5-3}{2}+\binom{5-2}{2}=4.$$

Mahadev 和 Roberts[2003]证明存在 G，L 使得 $I(G, L)/|V(G)|$ 任意接近于 1. 这一定理的解释是：存在某种情况，其中几乎每一个人都必须接受不在他们原来列表中的颜色. ■

尽管列表 $L(x)$可以在大小上发生变化，让我们考虑列表都是相同大小的情况. 这时，对于列表着色的一个重要问题是：给定图 G，寻找这样的最小的 c：使得无论把大小为 c 的什么列表分配给顶点，总可以找到一个列表着色. 如果一个图 G 对于任意大小为c 的列表，可以被 L 列表着色，我们说 G 是 c **可选的**(c-choosable). G 为 c 可选的最小 c 定义为 G 的**选择数**(choice number)，记作 $ch(G)$. 正如我们在图 3.45 中所看到的那样，图 $K_{3,3}$不是 2 可选的. 即使图 3.45b 中的列表分配可以产生一个列表着色，图 3.45a 中的列表分配不可能产生列表着色. 在练习 60 中，我们要求读者证明 $K_{3,3}$是 3 可选的. 因此，$ch(K_{3,3})=3$.

如我们所研究的很多其他图着色问题一样，计算一个图的选择数也是一个 NP 完全问题. Gravier[1996]证明了这一点. 但是，像其他着色问题一样，可以得到一般情况的选择数上界和特殊类图的精确值.

根据定义，如果 G 是 c 可选的，那么对于每一个顶点的任意 c 种颜色的列表分配存在列表着色. 特别地，如果我们把集合{1，2，…，c}分配给每一个顶点，列表着色(也必须是一个正确的图着色)一定存在. 至此，G 的色数至多是 c. 至此，我们证明了

$$ch(G)\geqslant\chi(G). \tag{3.10}$$

方程(3.10)中的等式有时候对某些图成立. 考虑有 n 个顶点的完全图 K_n. 回想一下, $X(K_n)=n$;每一顶点与另外一个顶点都是邻接的,因此需要 n 种颜色. K_n 的选择数 $ch(K_n)$ 是什么呢? 很容易看到 $ch(K_n)\leqslant n$. 因为当从一个顶点的列表中选出一个元素着色顶点时,不能选择的唯一的颜色是已分配给邻接顶点的颜色,而且有 $n-1$ 种这样的颜色. 由不等式 $ch(K_n)\leqslant n$ 及方程(3.10),有

$$ch(K_n) = n = \chi(K_n).$$

然而,我们当然不能总是得到方程(3.10)中的等式. 事实上, Erdös, Rubin 和 Taylor[1979]已证明选择数与色数之间的差可以任意大. 关于这一领域的进一步研究,可参见 Alon[1993]以及 Kratochvil, Tuza 和 Voigt[1999].

165

本节练习

1. 考虑下面纽约市西区的四个垃圾卡车巡回. 巡回 1 遍历从 21 街到 30 街的场所,巡回 2 遍历从第 28 街到 40 街的场所,巡回 3 遍历从第 35 街到第 50 街的场所,巡回 4 遍历从第 80 街到第 110 街的场所. 画出相应的巡回图.
2. 在练习 1 中,能够将每个巡回都安排在星期一和星期二,使得没有场所在同一天访问两次吗?
3. 对于图 3.47 中的每一个图:
   (a)确定它是否是 3 可着色的.
   (b)确定它的色数 $\chi(G)$.

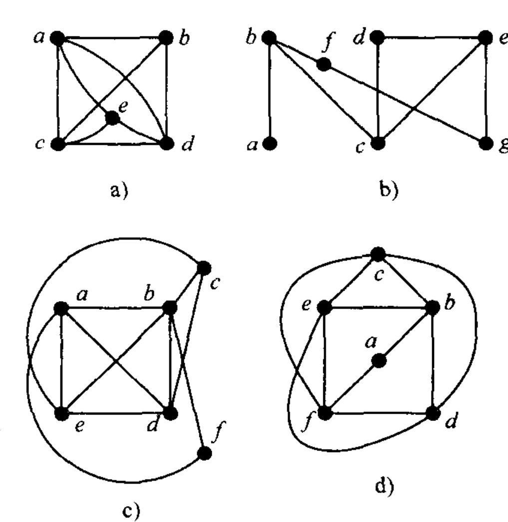


图 3.47 3.3 节练习中的图

4. 一个地方动物园想要游客参加动物喂养旅行,而且确定了下面的旅程. 旅程 1 观看狮子、大象和鸵鸟;旅程 2 观看猴子、鸟和鹿;旅程 3 观看大象、斑马和长颈鹿;旅程 4 观看鸟、爬行动物和熊;旅程 5 观看袋鼠、猴子和海豹. 如果动物一天的喂养不超过一次,只利用星期一、星期三和星期五能安排这些旅程吗?
5. 下面是正在考虑的纽约市垃圾卡车巡回(没有得到市长的支持). 巡回 1 在帝国大厦、麦迪逊广场花园和哈得逊河的 42 码头回收垃圾. 巡回 2 巡回格林威治村、42 码头、帝国大厦和大都会歌剧院. 巡回 3 访问谢伊球场、布朗克斯动物园和布鲁克林植物园. 巡回 4 到自由女神雕像和 42 码头;巡回 5 到自由女神雕像、纽约证券交易所和帝国大厦;巡回 6 访问谢伊球场、扬基棒球馆和布朗克斯动物园;巡回 7 访问纽约证券交易所、哥伦比亚大学和布朗克斯动物园. 假设环卫工人拒绝一周工作超过三天,能够分摊这些巡回,使得没有场所在给定的日子内被巡回超过一次吗?

166

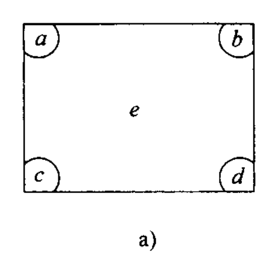
6. 下面的委员会需要调度会议时间.

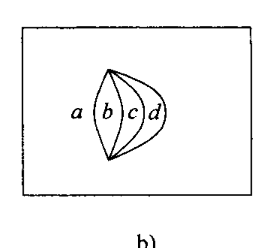
   $A=${斯密斯,约翰,布朗,格林}
   $B=${约翰,瓦格纳,蔡斯}
   $C=${哈里斯,奥利弗}
   $D=${哈里斯,约翰,梅森}
   $E=${奥利弗,卡明斯,拉森}

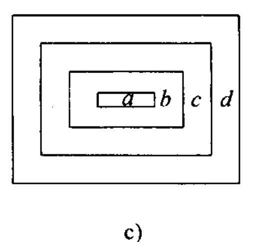
   三个会议时间足以调度这些委员会,使得没有成员同时出席两个会议吗? 为什么?
7. 给移动式无线电话机分配频率时,一个"环带"得到一个由那个环带的所有交通工具使用的频率. (因为相距太近或气象原因)相干扰的两个环带必须得到不同的频率. 如果有 6 个环带 a, b, c, d, e 和 f, 且 a 只干扰 b; b 干扰 a, c, d; c 干扰 b, d, e; d 干扰 b, c, e; e 干扰 c, d, f; f 只干扰 e, 那么需要多

少个不同的频率？

8. 给工人分配工作区域时，我们要确保如果两个工人工作相互干扰，那么他们将得到不同的工作区域. 如果有6名工人 a，b，c，d，e，f，且工人 a 干扰 b，e 和 f；工人 b 干扰 a，c 和 f；工人 c 干扰 b，d 和 f；工人 d 干扰 c，e 和 f；工人 e 干扰 a，d 和 f；工人 f 干扰所有其他工人，那么需要多少个工作区域？
9. 在一个程序的给定循环中有六个变量. 变量 A 必须在步骤1到步骤4存储，变量 B 在步骤3到步骤6存储，变量 C 在步骤4到步骤7存储，变量 D 在步骤6到步骤9存储，变量 E 在步骤8到步骤9存储，变量 F 在步骤9到步骤10存储. 为了存储需要多少个变址寄存器？
10. 寻找对应于图3.48中地图的图. 注意单一公共点不能作为公共边界.







a) b) c)

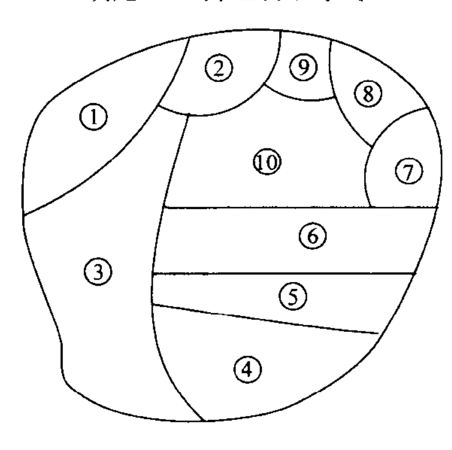
图3.48 地图

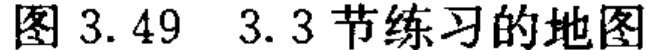
11. 把图3.49中的地图翻译成一个图 G 并计算 $\chi(G)$.
12. 对于图3.50中的每一个图：

167

(a)确定 $\omega(G)$. (b)确定 $\alpha(G)$.

(c)确定 $\chi(G)$ 并证明不等式(3.9).




图3.49 3.3节练习的地图

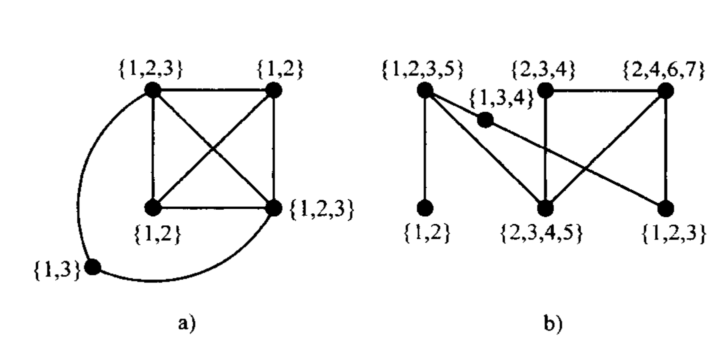


图3.50 练习的图

13. 设 G 是有17个顶点且色数是4的任意图.

(a)G 必须有一个大小为4的独立集合吗？

(b)G 必须有一个大小为5的独立集合吗？
14. 如果用4种颜色着色长度为11的回路 Z_{11} 的顶点，你能说出得到相同颜色的最大的顶点集合的大小吗？
15. 给出图 G 的例子，使得：

(a)$\chi(G)=\omega(G)$ (b)$\chi(G)>\omega(G)$
16. 设 $\theta(G)$ 是 G 的团的最小集合 S 的大小，使得每一个顶点在 S 的某个团中. $\theta(G)$ 和 $\chi(G)$ 之间的关系是什么？

168

17. 以 K_5 作为子图的任意图能是平面的吗？为什么？
18. 假设有四座房子和四种公共设施，每一座房子通过一条边与每一种公共设施相连接. 其结果图是平面的吗？

为什么？

19. K_6 能够来自一张地图吗？为什么？

20. (a)从 K_2 通过细分能够得到 K_3 吗？

(b)K_2 和 K_3 是同胚的吗？

21. 图 3.47 中的哪些图是平面的？

22. 使用下面的定理证明图 3.51a 中的图是非平面的：

(a)Kuratowski 定理(定理 3.2)　　(b)定理 3.3

23. 对于图 3.51b 中的图重复练习 22.

24. 对于图 3.47 中的所有图，寻找通过收缩边$\{c, e\}$所得到的图.

25. 对于图 3.47 中的所有图，寻找通过按下面的顺序收缩边$\{a, b\}$，$\{b, c\}$，$\{c, d\}$所得到的图.

26. 设顶点 u 在图 G 中的度 $\deg(u)$被定义为 u 的邻居数量. 设$\Delta(G)$是 G 的所有顶点的度的最大值. 证明$\chi(G)\leqslant 1+\Delta(G)$.

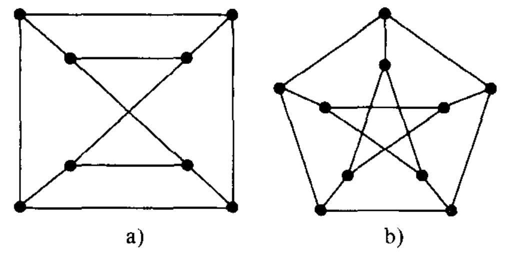


图 3.51　两个非平面图

27. 给出 G 的例子使得$\chi(G)<1+\Delta(G)$，其中$\Delta(G)$的定义同练习 26.

28. $\chi(G)=\omega(G)$的图 G 称为**弱 γ 完美的**(weakly γ-perfect). 给出一个弱 γ 完美图和一个非弱 γ 完美图的例子.

29. 证明每一个 2 可着色图都是弱 γ 完美的.

30. 关于图 3.23 中的图 c 和 g 说明算法 3.1.

31. 证明：根据算法 3.1，如果 $d(x, z)$是奇数则顶点 z 得到红色，如果 $d(x, z)$是偶数则顶点 z 得到蓝色，其中 x 是开始顶点.

32. 证明：如果一个图有一条奇数长度的闭链，那么它有一条奇数长度的回路.

33. 证明：$\chi(Z_p)=3$ 当且仅当 p 是奇数.

34. (a)使用 $T=\{0, 1, 2\}$确定 K_4 的 T 着色.

(b)使用 $T=\{0, 2, 4\}$确定 Z_6 的 T 着色.

169

(c)使用 $T=\{0, 1, 2, 4, 8, 15\}$寻找图 3.52 中的图的阶为 6 的 T 着色.

(d)使用 $T=\{0, 1, 3\}$寻找图 K_4 的 T 着色，且使得这一 T 着色的跨度为 7.

35. 当使用 T 集合$\{0, 1, 3, 4\}$时，确定 K_4 的 T 跨度.

36. 在确定一个高效 T 着色时，最小化阶和最小化跨度可能需要不同的 T 着色. 回想 $\chi(Z_5)=3$ 和根据定理 3.8，对于任意的 T 集合，有 $\chi_T(Z_5)=3$.

(a)对于 T 集合$\{0, 1, 4, 5\}$，使用 3 种颜色寻找 Z_5 的具有最小可能跨度的 T 着色.

(b)对于 T 集合$\{0, 1, 4, 5\}$，使用 4 种颜色寻找 Z_5 的具有最小可能跨度的 T 着色.

(c)对于 T 集合$\{0, 1, 4, 5\}$，使用 5 种颜色寻找 Z_5 的具有最小可能跨度的 T 着色.

(d)对于 T 集合$\{0, 1, 4, 5\}$，存在既最小化阶又最小化跨度的 Z_5 的一个 T 着色吗？

37. 证明：从使用颜色 1，2，…，j 的图 G 的一个顶点着色出发，通过颜色 $i(t+1)$取代颜色 i 可以得到一个 T 着色. 其中 t 表示这一给定的 T 集合中的最大颜色.

38. 证明：关于跨度，图 3.42 中的 T 着色是最高效的. 即证明：对于 $T=\{0, 3, 4, 7\}$，图 3.42 中的图不能使用颜色 1，2，…，nT 着色，其中 $n\leqslant 10$.

39. 寻找图 3.52 中的图的一个集合着色，对偶数下标的每一个顶点分配两种颜色，对奇数下标的每一个顶点分配三种颜色.

40. 使用 5 种颜色寻找 Z_4 的一个 2 元组着色.

41. 回想 $\chi_k(G)$是使得 G 有使用 m 种颜色的 k 元组着色的最小的 m. 求：

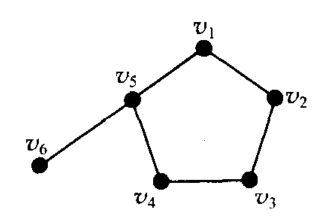


图 3.52　3.3 节练习所使用的图

(a)$\chi_2(Z_4)$　(b)$\chi_2(Z_3)$　(c)$\chi_2(K_4)$　(d)$\chi_3(Z_4)$　(e)$\chi_3(Z_3)$　(f)$\chi_3(K_4)$

42. 寻找图3.52中图G的$\chi_2(G)$.

43. 如果G是2可着色的且至少有一条边，证明$\chi_m(G)=2m$.

44. 证明$\chi_3(Z_5)=8$.

45. 对于图3.50中每一个图，确定它是否是带有给定列表分配的列表可着色. [170]

46. 证明：(G, L)对于某个p是p可加的当且仅当

$$|\bigcup\{L(x): x\in V\}|\geqslant\chi(G).$$

47. 图$K_{\binom{m}{2},\binom{m}{2}}$有各有$\binom{m}{2}$个顶点的两类顶点$A$和$B$，$A$中的每一个顶点$x$与$B$中的每一个顶点$y$邻接. 设$L$把集合$\{1, 2, \cdots, m\}$所有2元素子集给$A$的顶点，且对$B$中的顶点也类似. 确定

$$I(K_{\binom{m}{2},\binom{m}{2}}, L).$$

48. 设$K_{7,7}$的定义类似于例3.23中的$K_{10,10}$. 设对于所有的x，有$|L(x)|=3$及$|\bigcup L(x)|=6$，证明$K_{7,7}$关于L是1可加的.

49. 证明：

(a)Z_3不是2可选的.　(b)Z_3是3可选的.

50. (a)确定Z_4的选择数.　(b)对于偶数n，确定Z_n的选择数.

51. 假设给定有n个顶点的一个图的k元组着色. 如果我们把这一k元组着色看成是这个图的一个列表分配，那么可能有多少种不同的列表着色？

52. 一个图G是*k边可着色的*(*k*-edge-colorable)，如果你能够使用k种颜色着色边，使得具有公共顶点的两条边得到不同的颜色. 设*边色数*(edge chromatic number)$\chi'(G)$是使得G是k边可着色的最小的k. 陈述$\chi'(G)$与练习26中定义的数$\Delta(G)$之间的关系.（关于边着色的应用，参见Fiorini和Wilson[1977].）

53. 如果G有n个顶点，证明

$$\frac{n}{\alpha(G)}\leqslant\chi(G)\leqslant n-\alpha(G)+1.$$

54. 一个图G称为是*k临界的*(*k*-critical)，如果$\chi(G)=k$，但对于每一个顶点$u\in V(G)$有$\chi(G-u)<k$.

(a)寻找所有2临界图.

(b)给出一个3临界图的例子.

(c)你能找到所有的3临界图吗？

55. 如果$G=(V, E)$是一个图，它的补G^c是这样的图：它的顶点集合为V且对于V中的$x\neq y$，x和y之间有一条边，当且仅当$\{x, y\}\notin E$.

(a)陈述团数$\omega(G)$和顶点独立数$\alpha(G^c)$之间的关系.

(b)回想$\theta(G)$是覆盖G的所有顶点的最小的团数. 证明$\chi(G)=\theta(G^c)$.

(c)我们说G是*弱α完美的*(weakly α-perfect)，如果$\theta(G)=\alpha(G)$. 给出一个弱α完美图及一个非弱α完美图的例子. [171]

56. G*称为是γ完美的*(γ-perfect)*或α完美的*(α-perfect)，如果G的每一个生成子图都是弱γ完美的或弱α完美的. 分别给出下列各图的例子：

(a)γ完美　(b)弱γ完美但非γ完美

(c)α完美　(d)弱α完美但非α完美

57. Lovász[1972a, b]证明：一个图G是γ完美的，当且仅当它是α完美的. 因此，γ完美(或α完美)的图称为*完美*(perfect)的. 关于完美图的更多内容和它们的应用可参见Golumbic[1980].

(a)证明：G是弱γ完美的当且仅当G是弱α完美的不成立.

(b)证明：G是γ完美的当且仅当G^c是γ完美的(可以使用Lovasz的结果).

58. (Tutte[1954]、Kelly和Kelly[1954]、Zykov[1949])证明：对于任意整数$k>1$，存在一个图G使得

$\omega(G)=2$ 且 $\chi(G)=k$.

59. 证明定理 3.6.

60. 图 3.45a 给出了 $K_{3,3}$的一个列表分配，表明 $K_{3,3}$不是 2 可选的. 这一练习给出 $K_{3,3}$是 3 可选的证明. 设 L 是 $K_{3,3}$的一个列表分配，其中每一个列表的大小都为 3.

(a)假设两个非邻接顶点的列表分享同一种颜色. 通过使用这一相同颜色证明存在一个 L 列表着色.

(b)假设非邻接顶点的每一对列表都不分享同一种颜色. 证明存在一个 L 列表着色.

(c)证明为什么 $ch(K_{3,3})=3$.

3.4 色多项式⊖

3.4.1 定义和例子

假设 G 是一个图，且 $P(G, x)$计数以至多 x 种颜色着色 G 的方法数量. 计数给地图着色的方法数量的相关思想(参见练习 3)是由 Birkhoff[1912]在研究四色猜想时提出的，我们将在下面讨论. 数 $P(G, x)$是由 Birkhoff 和 Lewis[1946]提出的. 注意，如果使用 x 种颜色不能着色 G 则 $P(G, x)$是 0. 现在，$\chi(G)$是使 $P(G, x)\neq 0$ 的最小正整数. 研究 $P(G, x)$的最主要的原因之一是想了解 $\chi(G)$的有关情况. 本节将较为详细地研究数 $P(G, x)$，要频繁使用第 2 章的计数技术.

172

首先考虑图 3.53 所示的图 K_2. 如果 x 种颜色可用，那么它们中的任意一种都能用于着色顶点 a，其余的 $x-1$ 种颜色的任意一种都能用于着色顶点 b，因此，根据乘法规则，

$$P(K_2,x) = x(x-1) = x^2 - x.$$

特别地，

$$P(K_2,4) = 16 - 4 = 12.$$

图 3.53 图 K_2

至多用 4 种颜色着色 K_2 的 12 种方法如表 3.4 所示.

表 3.4 使用红色(R)、绿色(G)、蓝色(B)和黄色(Y)对图 3.53 中的图 K_2 的着色

a	R	R	R	G	G	G	B	B	B	Y	Y	Y
b	G	B	Y	R	B	Y	R	G	Y	R	G	B

下面考虑图 3.54 中的图 K_4. 如果有 x 种颜色可用，对顶点 a 的着色就存在 x 种选择. 对于其中每一种选择，对于顶点 b 存在 $x-1$ 种选择，因为 b 有到 a 的一条边；对于其中的每一种选择，对于顶点 c 存在 $x-2$ 种选择，因为 c 有到 a 和 b 的边；对于其中的每一种选择，对于顶点 d 存在 $x-3$ 选择，因为 d 有到 a，b，c 中每一个的边. 因此，有

$$P(K_4,x) = x(x-1)(x-2)(x-3) = x^4 - 6x^3 + 11x^2 - 6x.$$

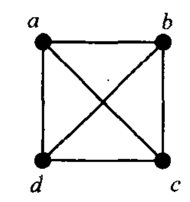


图 3.54 图 K_4

为了使用 x 种或更少的颜色着色图 3.55 中的图，对于顶点 f 存在 x 种选择. 那么对于顶点 a 存在剩余的 $x-1$ 种选择，对于其中的每一种选择，对于顶点 b 也存在 $x-1$ 种选择(因为 b 可以得到与 a 相同的颜色)，对于顶点 c 也存在 $x-1$ 种选择，对于顶点 d 也存在 $x-1$ 种选择，对于顶点 e 也存在 $x-1$ 种选择. 因此，

$$P(G,x) = x(x-1)^5 = x^6 - 5x^5 + 10x^4 - 10x^3 + 5x^2 - x.$$

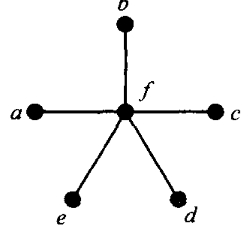


图 3.55 一个图

现在研究图 3.56 中的图 Z_4. 这里，a 和 b 必须得到不同的颜色，但

⊖ 除 7.16 节之外，后面的内容都不需要本节的内容. 但强烈地鼓励读者学习本节内容，因为它提供了很多第 2 章讲述的计数技术的应用.

是 a 和 c 可以得到相同的颜色．类似地，b 和 d 可以得到相同的颜色，等等．如果 x 种颜色可用，那么对于顶点 a 存在 x 种选择．这时，b 和 d 或者得到相同的颜色或者不同的颜色．分两种情况讨论更加方便．

173

情况 1：b 和 d 得到相同的颜色．

情况 2：b 和 d 得到不同的颜色．

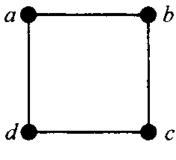


图 3.56 图 Z_4

在情况 1 中，c 可以得到不用于 b 和 d 的任意一种颜色．因此，对于顶点 a 存在 x 种选择，$x-1$ 种选择的每一种选择都可共用于 b 和 d，而 $x-1$ 种选择的每一种选择可用于 c．因此，使用 x 种或更少的颜色且 b 和 d 得到相同颜色的着色数量是

$$x(x-1)^2.$$

在情况 2 中，对于顶点 a 存在 x 种选择，对于顶点 b 存在 $x-1$ 种选择，对于顶点 d 存在 $x-2$ 种选择(因为它得到的颜色必须不同于 b 所得到的颜色)，于是对于顶点 c 存在 $x-2$ 种选择(因为它不可能得到用于 b 和 d 的颜色的任何一种，但是它可以得到 a 的颜色)．因此，b 和 d 得到不同颜色的着色数量是

$$x(x-1)(x-2)^2.$$

因为或者情况 1 或者情况 2 成立，加法规则给出

$$P(Z_4,x) = x(x-1)^2 + x(x-1)(x-2)^2 = x^4 - 4x^3 + 6x^2 - 3x. \quad (3.11)$$

读者应该注意到，在我们给出的每一个例子中，$P(G, x)$都是 x 的多项式．这总成立．因此，称 $P(G, x)$为色多项式(chromatic polynomial)是有意义的．

定理 3.8 *$P(G, x)$总是一个多项式*[⊖]．

我们将在下面证明这一定理．回想 $\chi(G)$是使 $P(G, x)\neq 0$ 的最小正整数，也就是说，这样的 x 不是多项式 $P(G, x)$的根．因此，色数可以通过求一个多项式的根计算．Birkhoff 研究四色问题的方法是基于尝试刻画色多项式是什么多项式的思想，特别是地图(或平面图)的色多项式，然后检查 4 是否是这些多项式的任意一个的根．到今天，刻画色多项式的问题仍没有得到解决．在下面我们将讨论这一问题．

174

【例 3.24 立法委员会会议调度(再探例 3.11)】 使用三种或少于三种颜色计数图 3.28中的图 G 的着色数．我们首先计算 $P(G, x)$．如果对于教育委员会的着色存在 x 种选择，那么对于住房委员会的着色存在 $x-1$ 种选择，对于健康委员会的着色存在 $x-2$ 种选择．留给运输委员会的着色是 $x-2$ 种选择，对于环境委员会的着色是 $x-2$ 种选择，最后对于财政委员会的着色有 $x-1$ 种选择．因此，有

$$P(G,x) = x(x-1)^2(x-2)^3$$

以及

$$P(G,3) = 12.$$

这一结果与例 1.4 的讨论结论一致．在表 1.7 中我们描述了以三种或少于三种颜色着色的 12 种可能情况． ■

接下来，我们给出关于色多项式的简单但又很基础的两个结果．其一是关于有 n 个顶点但没有边的空图(empty graph) I_n．

定理 3.9 (a)*如果 G 是 K_n，那么*

$$P(G,x) = x(x-1)(x-2)\cdots(x-n+1). \quad (3.12)$$

⊖ 这一定理是由 Birkhoff[1912]对于地图所发现的．对于任意图，它是由 Birkhoff 和 Lewis[1946]提出的．

(b)如果 G 是 I_n，那么 $P(G,\ x)=x^n$.

证明　(a)对于第一个顶点的着色存在 x 种选择，对于第二个顶点的着色存在 $x-1$ 种选择，以此类推.

(b)对于 n 个顶点的每一个顶点的着色都存在 x 种选择. ■

(3.12)式的右边的表达式将频繁出现，所以为方便起见给它起一个名字，称它为 $x^{(n)}$.

3.4.2　化简定理

组合数学中一个很常用的技术是把较大的计算化简成一些较小的计算. 本书中，我们经常使用这一技术，特别是在第6章，我们要研究递推关系并把 x_n 的计算化简成对于小于 n 的 k 求 x_k 的值. 对于色多项式的计算，事实证明这是一个非常有用的技术. 本小节开发和运用几个化简定理，这些定理可以用于把任意色多项式的计算化简成为带有较少边的图的色多项式计算，直至完全图或空图的色多项式的计算.

现在描述第一个定理.

定理 3.10(两部分定理)　*假设我们把 G 的顶点集合划分成互不相交的集合 W_1 和 W_2，且设 G_1 和 G_2 分别是由 W_1 和 W_2 生成的子图. 假设在 G 中，没有把 W_1 的顶点与 W_2 的顶点连接起来的边. 于是，有*

175

$$P(G,x)=P(G_1,x)P(G_2,x).$$

证明　如果有 x 种颜色可用，G_1 的着色就存在 $P(G_1,\ x)$种选择；对于其中每一种选择，G_2 的着色存在 $P(G_2,\ x)$种选择. 这是因为 G_1 的着色不影响 G_2 的着色，因为不存在连接两个部分的边，所以根据乘法规则这一定理成立. ■

为了说明这一定理，考虑图 3.57 中的图 G，这时有

$$P(G,x)=(x^{(3)})(x^{(3)})=(x^{(3)})^2=[x(x-1)(x-2)]^2.$$

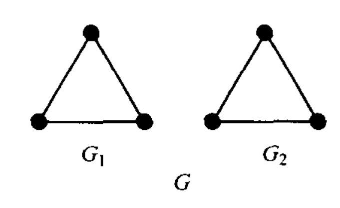


图 3.57　有两个部分的图

为了陈述下一个关键的化简定理，假设 α 是图 G 的一条边，连接顶点 a 和 b. 我们从 G 出发定义两个新图. 图 G'_α 是通过删除边 α 但保留顶点 a 和 b 而得到的. 图 G''_α 是通过把顶点 a 和 b 看成相同的顶点而得到的. 在这一情况下，这一新的组合顶点与所有与 a 或 b 连接过的顶点相连接. (如果 a 和 b 都与一个顶点 c 连接，那么只包含从这一组合顶点发出的一条边. 在 3.3.2 节中我们说 G''_α 是从 G 通过收缩边 α 得到的.)图 3.58 和图 3.59 给出这两个新图.

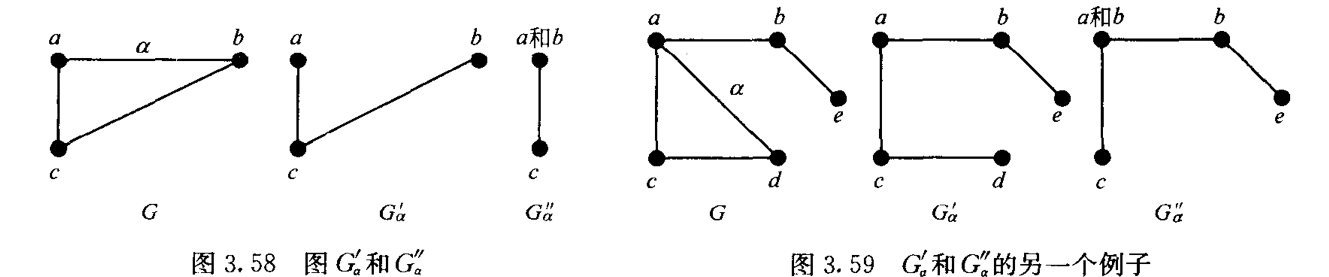


图 3.58　图 G'_α 和 G''_α

图 3.59　G'_α 和 G''_α 的另一个例子

定理 3.11(基础化简定理㊀)

$$P(G,x)=P(G'_\alpha,x)-P(G''_\alpha,x).\tag{3.13}$$

证明　假设我们用至多 x 种颜色着色 G'_α，其中边 α 在 G 中连接顶点 a 和 b. 那么或者 a 和 b

㊀ 这一定理是由 Birkhoff 和 Lewis[1946]提出的.

得到不同的颜色，或者 a 和 b 得到相同的颜色．使得 a 和 b 得到不同颜色的着色 G'_α 的方法数量与着色 G 的方法数量相同，即为 $P(G, x)$．使得 a 和 b 得到相同颜色的着色 G'_α 的方法数量与着色 G''_α 的方法数量相同，即 $P(G''_\alpha, x)$．因为我们知道在 G''_α 中，a 和 b 会得到相同的颜色，另外迫使联合顶点 a 和 b 得到与指定给顶点 c 的颜色不同的颜色，当且仅当 a 和 b 中的一个或二者被迫得到与指定给顶点 c 的颜色不同的颜色．根据加法规则，这一结果满足 [176]

$$P(G'_\alpha,x) = P(G,x) + P(G''_\alpha,x).$$ ■

为了说明这一定理，考虑图 3.60 中的图 G 并使用基础化简定理计算它的色多项式．我们选择顶点 1 和顶点 3 之间的边作为 α，所以得到图中所示的 G'_α 和 G''_α．现在，图 3.60 的图 G''_α 是完全图 K_2，因此，根据定理 3.9，我们知道

$$P(G''_\alpha,x) = x(x-1). \tag{3.14}$$

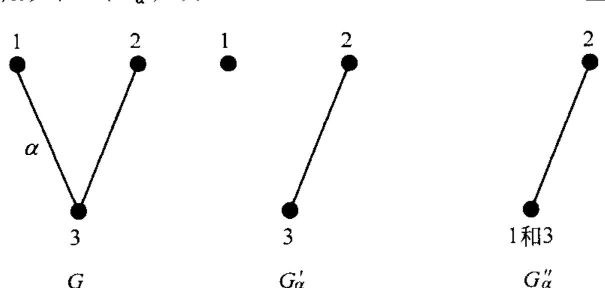


图 3.60　基础化简定理的运用

图 G'_α 有两个部分 K_1 和 K_2．根据两部分定理，有

$$P(G'_\alpha,x) = P(K_1,x)P(K_2,x). \tag{3.15}$$

根据定理 3.9，(3.15)式右边的第一个表达式是 x，而第二个表达式是 $x(x-1)$．因此，有

$$P(G'_\alpha,x) = x \cdot x(x-1) = x^2(x-1). \tag{3.16}$$

把(3.14)式和(3.16)式带入(3.13)式，我们得到

$$\begin{aligned} P(G,x) &= x^2(x-1) - x(x-1) \\ &= x(x-1)(x-1) \\ &= x(x-1)^2. \end{aligned}$$

当然，这一表达式可以直接得到．然而，这是使用基础化简定理的一个很好的说明．顺便提一下，对 $G''_\alpha = K_2$ 再一次运用基础化简定理，易证

$$P(G''_\alpha,x) = P(K_2,x) = P(I_2,x) - P(I_1,x), \tag{3.17}$$ [177]

同样，因为 $K_1 = I_1$，由(3.15)式和(3.17)式，我们有

$$P(G'_\alpha,x) = P(I_1,x)[P(I_2,x) - P(I_1,x)]. \tag{3.18}$$

最后，把(3.17)式和(3.18)式代入到(3.13)式得到

$$P(G,x) = P(I_1,x)[P(I_2,x) - P(I_1,x)] - [P(I_2,x) - P(I_1,x)]. \tag{3.19}$$

我们已把 $P(G, x)$ 简化成表达式(3.19)，而这一表达式只需要知道不同 k 值的多项式 $P(I_k, x)$．

作为第二个例子，我们使用基础化简定理计算 $P(K_3, x)$．考虑图 3.61，如果 $H = K_3$，那么 H'_α 是图 3.60 中的图 G，且 H''_α 是 K_2．因此，有

$$\begin{aligned} P(K_3,x) &= P(G,x) - P(K_2,x) \\ &= x(x-1)^2 - x(x-1), \end{aligned}$$

图 3.61　基础化简定理的第二个应用

其中第一个表达式来自于我们前面的计算，而第二个表达式来自于 $P(K_2, x)$ 的公式．简化后，我们得到

$$\begin{aligned} P(K_3,x) &= x(x-1)[(x-1)-1] \\ &= x(x-1)(x-2) \\ &= x^{(3)}, \end{aligned}$$

这一表达式与定理 3.9 一致．

读者应该注意到，基础化简定理的每一次运用都减少留下的每个图中的边的数量．因此，通

过反复使用基础化简定理，最终我们一定结束于没有边的图，即形如 I_k 的图. 我们的第一个例子说明了这一点. 在任何情况下，实际上这表明定理 3.8 一定成立，即 $P(G, x)$ 总是 x 的多项式. 因为有 $P(I_k, x)=x^k$. 因此，我们最终把 $P(G, x)$ 简化成每一项是形如 x^k 的和、差或积的表达式(可以通过对图中的边数作归纳而得到形式证明).

178

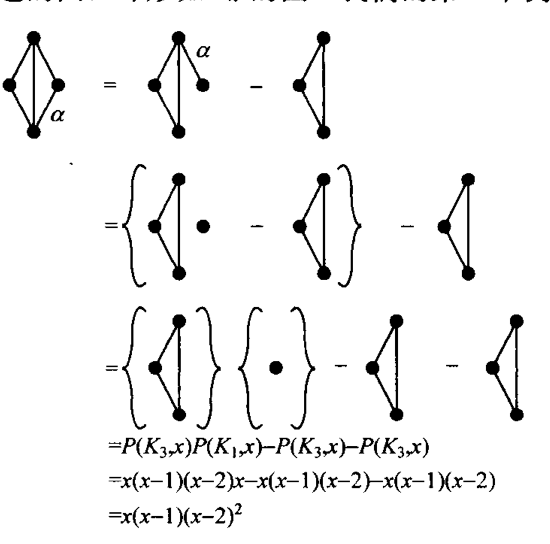


图 3.62 基础化简定理的另一个应用

图 3.62 给出基础化简定理的最后说明. 在这个图中，我们简单地画出一个图来表示这个图的色多项式. 每一步都指出了边 α.

3.4.3 色多项式的性质[⊖]

我们已经指出，Birkhoff 引入色多项式的愿望之一是说明什么样的多项式是色多项式，从而研究那些色多项式的根. 本节研究色多项式的性质. 我们将发现通过研究图的色多项式可以对一个图有很多了解. 练习 14 到练习 17 给出这里所陈述的定理的证明框架，并进一步给出色多项式的性质.

第一个定理概括了色多项式的基本性质. 这些性质可以通过 3.4.1 节和 3.4.2 节的所有例子得到证实.

定理 3.12(Read[1968]) *假设 G 是有 n 个顶点的图，且*

$$P(G,x)=a_px^p+a_{p-1}x^{p-1}+\cdots+a_1x+a_0.$$

那么

(a) $P(G, x)$ 的度是 n，即 $p=n$.

(b) x^n 的系数是 1，即 $a_n=1$.

(c) 常数项等于 0，即 $a_0=0$.

(d) 或者 $P(G, x)=x^n$ 或者 $P(G, x)$ 的系数和等于 0.

定理 3.13(Whitney[1932]) *$P(G, x)$ 是 x 的连续幂的和，且这些幂的系数的符号交替. 即对于某个 I，有*

$$P(G,x)=x^n-\alpha_{n-1}x^{n-1}+\alpha_{n-2}x^{n-2}\mp\cdots\pm\alpha_0, \tag{3.20}$$

179

且对于 $i\geqslant I$，$\alpha_i>0$，对于 $i<I$，$\alpha_i=0$.

定理 3.14(Read[1968]) *在 $P(G, x)$ 中，x^{n-1} 的系数的绝对值是 G 的边的数量.*

遗憾的是，我们在定理 3.12 和 3.13 中列出的色多项式的性质不能刻画色多项式. 有很多满足所有这些条件的多项式 $P(x)$，但是它们不是任何图的色多项式. 例如，考虑下面的多项式：

$$P(x)=x^4-4x^3+3x^2.$$

注意 x^n 的系数是 1，常数项是 0，系数和是 0，且从 x^1 开始系数符号交替，直到系数为零. 然而，$P(x)$ 不是任何图的色多项式. 如果它是的话，根据定理 3.12(a)，这个图的顶点数量应该等于 4. 根据定理 3.14，边数也应该等于 4. 不存在带有四个顶点和四条边的图以这一多项式作为色多项式，这很容易证实. 也可以参见 Lehmer[1985]，他找到了关于 6 个顶点或少于 6 个顶点的所有图的色多项式.

关于色多项式的更多结果，可以参见 Jensen 和 Toft[1995]、Liu[1972]和 Read[1968].

⊖ 本小节可以跳过.

本节练习

1. 确定图 3.63 中的每一个图的色多项式. [180]
2. 对于图 3.63 中的每一个图，确定以最多三种颜色着色这个图的方法数量.
3. 一张地图 M 的色多项式 $P(M, x)$ 是使用 x 种或少于 x 种颜色着色 M 的方法数量. 确定图 3.64 中的地图的 $P(M, x)$.
4. 设 L_n 是由 n 个顶点的简单链组成的图. 确定 $P(L_n, x)$ 的公式.

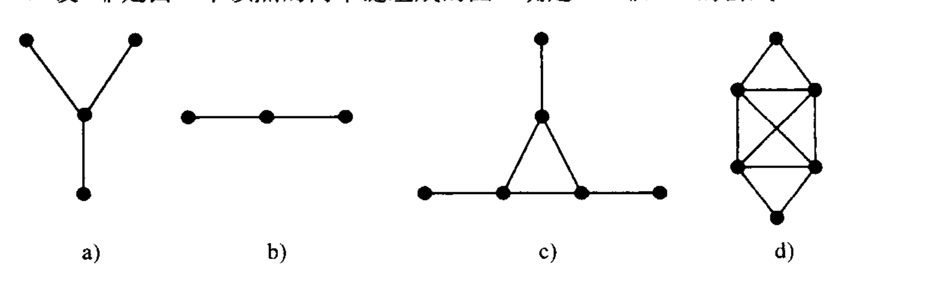


图 3.63　3.4 节练习的图

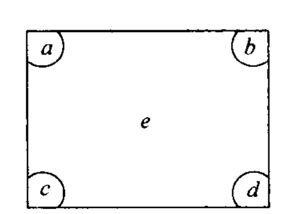


图 3.64　3.4 节练习的地图

5. 对于图 3.65 中的每一个图，使用化简定理确定其色多项式.（你可以把各图简化成有前面已知色多项式的图.）

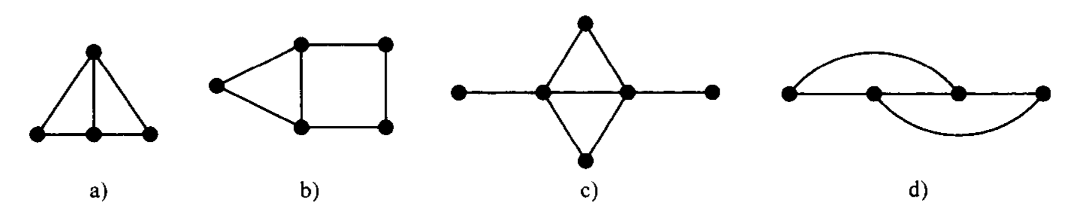


图 3.65　3.4 节练习的图

6. 如果 G 是图 3.65a 的图，对于各种 k 使用多项式 $P(I_k, x)$ 表示 $P(G, x)$.
7. 如果 L_n 如练习 4 中的定义，那么 $P(Z_n, x)$，$P(Z_{n-1}, x)$ 和 $P(L_n, x)$ 之间的关系是什么?
8. 使用化简定理寻找图 3.66 中的地图的色多项式(参见练习 3). 你可以使用练习 4 的结果.
9. 设 $N(G, x)$ 是正好使用 x 种颜色着色 G 的方法数量. 对于下面每一个图以及给定的 x 的值，确定 $N(G, x)$.
   (a) Z_5，$x=4$　　(b) K_5，$x=6$　　(c) L_5，$x=3$
10. 对于 $r \leqslant x$，利用数 $N(G, x)$ 确定 $P(G, x)$ 的表达式. [181]
11. 如果我们有图 G 的某些顶点的一个着色，我们称这是 G 的一个**子着色**(subcoloring). 与 G 的某些顶点的子着色一致的 G 的所有顶点的着色称为这一子着色的**扩展**(extension). 图 3.67 给出一个图 G 和 G 的三个子着色. 如果正好只有一种额外的颜色可用，比如说是蓝色，那么第一个子着色可以正好用一种方法扩展到 G，即通过把顶点 a 着色为蓝色，顶点 b 着色为红色. 然而，第二个子着色可以用两种方法扩展到 G，即通过着色 a 为蓝色，着色 b 为红色，或着色 a 为红色，着色 b 为蓝色.

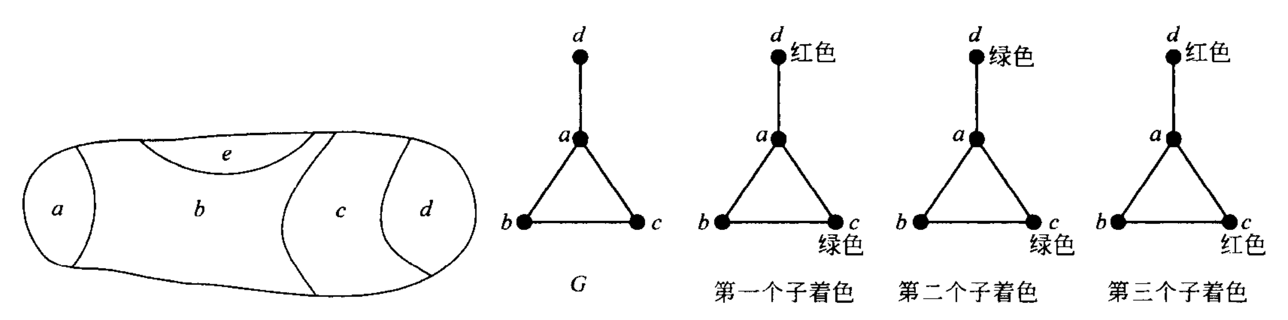


图 3.66　3.4 节练习的地图

图 3.67　一个图 G 和 G 的三个子着色

(a)图 3.67 中的第三个子着色有多少种扩展?

(b)考虑图 3.68 中的图 G 和图中所示的顶点 a，b 和 c 的子着色. 如果只有颜色绿、红、蓝和棕色是可用的，那么这一子着色到整个 G 的扩展有多少种?

(c)考虑图 3.69 中的图和此图所示的子着色. 如果红，蓝和绿这三种颜色可用，使用三种或少于三种颜色将子着色扩展到整个图的着色，确定每一个子着色的扩展数量.

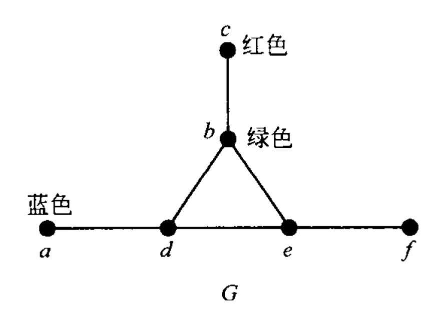
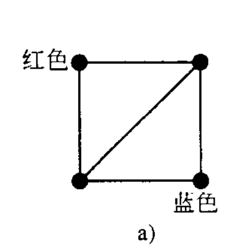
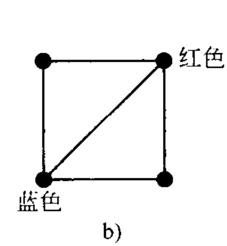


图 3.68 一个图及一个子着色





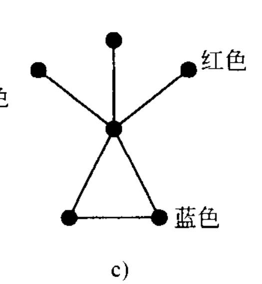


图 3.69 图和子着色

182 ~ 183

12. 重复练习 11(c)，至多使用 $x(x\geqslant 3)$ 种颜色，确定扩展数量.

13. 证明下列多项式不是色多项式.

(a)$P(x)=x^8-1$　　(b)$P(x)=x^5-x^3+2x$

(c)$P(x)=2x^3-3x^2$　　(d)$P(x)=x^3+x^2+x$

(e)$P(x)=x^3-x^2+x$　　(f)$P(x)=x^4-3x^3+3x^2$

(g)$P(x)=x^9+x^8-x^7-x^6$

14. 证明定理 3.12 的(c)和(d).

15. 通过对边数 e 作归纳，并运用基础化简定理证明定理 3.12 的(a)和(b).

16. 通过对边数 e 作归纳，并运用基础化简定理证明定理 3.13.

17. 通过对边数 e 作归纳，根据定理 3.13 证明定理 3.14.

18. 如果 G 是有 n 个顶点的连通图，证明：对任意正整数 q，有 $P(G, q)\leqslant q(q-1)^{n-1}$.

19. 证明：对于任意的 $\lambda<0$，有 $P(G, \lambda)\neq 0$.

20. (a)如果 G 有 k 个连通分支，证明使 x^i 在 $P(G, x)$ 有非零系数的最小 i 至少是 k.

(b)这一最小 i 一定要等于 k 吗? 为什么?

(c)证明：如果 $P(G, x)=x(x-1)^{n-1}$，那么 G 是连通的.

21. 假设 W_n 是 $n+1$ 个顶点的轮子(wheel)，即从 Z_n 出发，通过增加一个顶点并把这个顶点与 Z_n 的所有顶点连接得到的图. W_4 和 W_5 如图 3.70 所示. 确定 $P(W_n, x)$. 可以使用 $P(Z_n, x)$ 写出你的答案.

22. 设 Z_n 是长度为 n 的回路.

(a)证明：对于 $n\geqslant 3$，$(-1)^n[P(Z_n, x)-(x-1)^n]$ 是一个独立于 n 的常数.

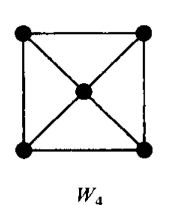
(b)通过计算(a)中的常数，求解 $P(Z_n, x)$.

23. 假设 H 是 G 的一个团，且我们有两个至多使用 x 种颜色的 H 的不同子着色. 证明：使用至多 x 种颜色扩展到 G 着色的数量对于每一个子着色都是相同的.

184

24. 下面是另一个化简定理. 假设 H 和 K 是 G 的生成子图，且有 $V(G)=V(H)\cup V(K)$ 和 $E(G)=E(H)\cup E(K)$，而且假设 $V(H)\cap V(K)$ 是有 p 个顶点的图 G 的一个团. 那么

$$P(G,x)=\frac{P(H,x)P(K,x)}{x^{(p)}}.$$



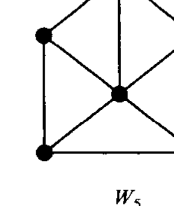


图 3.70 轮子 W_4 和 W_5

(a)如果 H 是由$\{a, d, e, f\}$所生成的子图，K 是由$\{c, b, d, e\}$所生成的子图，对于图 3.68 中的图 G 说明此结果(不考虑

子着色).

(b)利用练习23的结果证明此定理.

25. 如果色多项式 $P(K_n, x)$ 被展开，x^k 的系数被记为 $s(n, k)$，并称其为第一类斯特林数(Stirling number of the first kind). 练习25到练习27都将探究此数. 确定：

(a) $s(n, 0)$　　(b) $s(n, n)$　　(c) $s(n, 1)$　　(d) $s(n, n-1)$

26. 证明：

$$s(n,k) = (n-1)s(n-1,k) + s(n-1,k-1).$$

27. 使用练习26的结果说明如何通过类似于帕斯卡三角形的方法计算第一类斯特林数，并运用你的想法计算 $s(6, 3)$.

3.5 树

3.5.1 树的定义和例子

本节和下一节考虑图论中最有用的一个概念，那就是树的概念. 树(tree)是一个连通的且没有回路的图 T. 图3.71给出一些树.

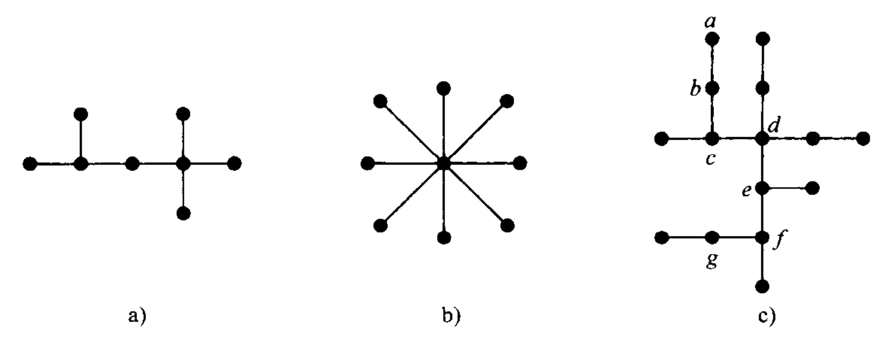


图3.71 一些树

185

【例3.25 电话树】 很多公司及其他机构都预先安排电话链以便在诸如使公司无法开张的暴风雪等紧急情况下通知他们的雇员. 在这样的电话链中，一位主管做出一个决定(例如，由于大雪而关闭公司)并给几个指定的人打电话，而这些人再分别给几个指定的人打电话，以此类推. 我们设这一公司的人是一个图的顶点，如果 a 呼叫 b 则包含一条从 a 到 b 的边(虽然打电话不是对称的，但我们使用无向的边). 这一结果图就是诸如图3.72所示的树.

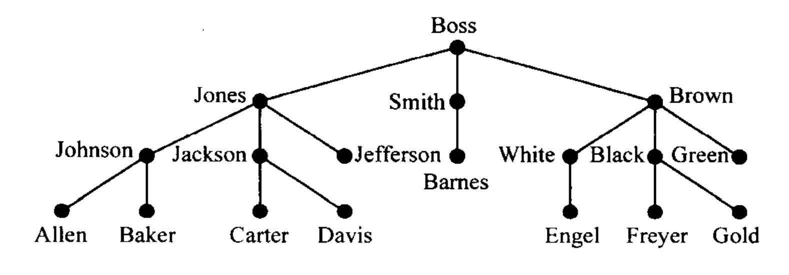


图3.72 电话链 ■

【例3.26 邮件分类⊖】 在美国如果发送人遵守邮电服务指南，打算寄出的邮件都会书写一个由特定几个十进制数字组成的ZIP代码. 到达邮局的邮件首先根据最有意义的数字被分类成10堆. 然后，每一堆根据下一个最有意义数字再被分成10堆. 以此类推. 这一分类过程可以由

⊖ 这一例子基于 Deo[1974]的工作.

一棵树概括，图 3.73 给出了其中的一部分. 为了给出更简单的例子，假设我们通过给邮件一个仅由三个数字组成的代码而不是 ZIP 代码，每一个数字是 0 或 1，在一个大的机构内分类邮件. 于是分类树如图 3.74 所示.

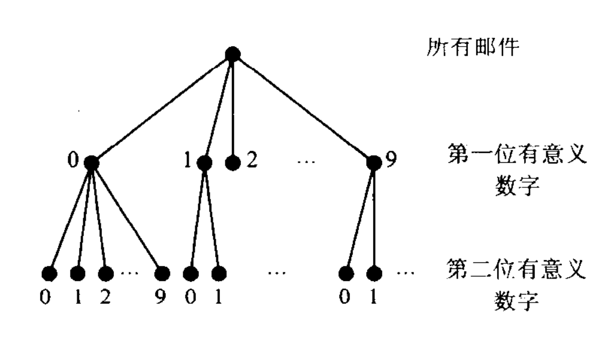


图 3.73　通过 ZIP 代码分类邮件的分类树的一部分

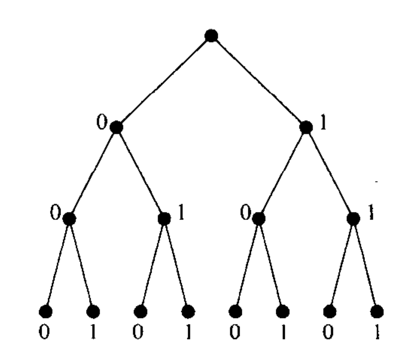


图 3.74　由三个二进制数字编码的邮件分类树 ■

186

【例 3.27 进化树】 现代生物学中的广泛接受的观点是，所有现存的有机体都来自于一个共同的祖先，而且当给定群落通过某个过程，如变异，分裂成两个或更多的群落时，新的物种产生. 通常使用树模型化物种进化成新物种的进化过程. 树的顶点是物种，从一个物种出发的一条边延伸到一个直接后代. 图 3.75 给出一个进化树. 进化树也出现在其他非生物学的领域. 例如，在语言学中，我们通过拼写中的"变异"研究单词随时间的进化. 研究进化的科学领域种系发展史中的一个共同问题是，从现存物种的信息重新构造进化树或*种系发生树*(phylogenetic tree). 对此我们将在 3.5.7 节中更详细说明. ■

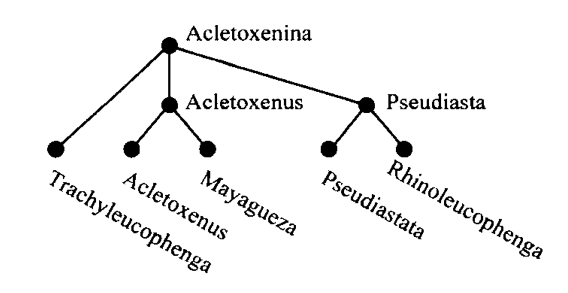


图 3.75　果蝇科的种系发生树(部分)(双翅鱼)(基于 Grimaldi[1990]的数据)

本节和下一节考虑树的其他多种应用，特别地，强调在有机化学、种系发生树重构和在计算机科学中的搜索与排序问题上的应用.

187

3.5.2　树的性质

树的一个基本性质是通过关注树的顶点数量和边的数量之间的关系而得到的.

定理 3.15　*如果 T 是一棵有 n 个顶点和 e 条边的树，那么 $n=e+1$.*

我们已画出的任意一棵树都可以说明定理 3.15. 我们将在 3.5.3 节中给出证明.

注意性质 $n=e+1$ 不能刻画树. 存在这样的图，它有性质 $n=e+1$，但不是树(练习 6). 然而，我们有下面的结果，这一结果将在 3.5.5 节中给出证明.

定理 3.16　*假设 G 是一个有 n 个顶点和 e 条边的图. 那么 G 是树当且仅当 G 是连通的且 $n=e+1$.*

接下来我们注意到定理 3.15 的一个有趣的推论. 这一结果用于计数树的数目.

定理 3.17　*如果 T 是一棵有多于 1 个顶点的树，那么至少存在两个度为 1 的顶点.*

证明　因为 T 是连通的，所以每一个顶点的度一定≥ 1(为什么). 现在根据定理 3.15 和 3.1，有$\sum \deg(u)=2e=2n-2$. 如果 $n-1$ 个顶点的度≥ 2，那么这些度的和必定至少是 $2(n-1)+1=2n-1$，这一结果大于 $2n-2$. 因此，不超过 $n-2$ 个顶点的度≥ 2. ■

3.5.3 定理 3.15 的证明[㊀]

定理 3.18 在一棵树 T 中，如果 x 和 y 是任意两个顶点，那么存在一条且仅存在一条连接 x 和 y 的简单链.

证明 我们知道，因为 T 是连通的，所以 x 和 y 之间存在一条链. 回想一下一条简单链是没有重复顶点的. x 和 y 之间的最短链一定是一条简单链. 因为如果 x，u，…，v，…，v，…，w，y 是 x 和 y 之间的一条最短的链，且有重复顶点 v，那么我们略去重复顶点 v 之间的部分，得到从 x 到 y 的一条更短的链. 因此，存在一条连接 x 和 y 的简单链. 接下来假设 $C_1=x$，x_1，x_2，…，x_k，y 是连接 x 和 y 的最短的链. 我们证明不可能存在其他连接 x 和 y 的简单链. 假设 C_2 是这样的一条链. 设 x_{p+1} 是使 C_1 和 C_2 不同的 C_1 中的第一个顶点，设 x_q 是使 C_1 和 C_2 相同的 C_1 中 x_p 后面的下一个顶点. 那么我们得到一条回路：沿着 C_1 从 x_p 到 x_q，然后沿着 C_2 从 x_q 返回到 x_p，这与 T 是一棵树相矛盾. ■ 188

为了说明这一定理，我们注意到，在图 3.71 的树 c 中，连接顶点 a 和 g 的唯一简单链由 a，b，c，d，e，f，g 给出. 现在我们返回到定理 3.15 的证明.

定理 3.15 的证明 对 n 作归纳进行这一证明. 如果 $n=1$，结果是显然的. 只有一个顶点的树没有边. 现在假设对于顶点数量少于 n 的所有树这一结果成立，并假设树 T 有 n 个顶点. 在 T 中挑选出任意一条边 $\{u, v\}$. 根据定理 3.18，u，v 是 u 和 v 之间的唯一简单链. 如果我们从 G 中把边 $\{u, v\}$ 去除(但是留下顶点 u，v)，我们得到一个新图 H. 现在在 H 中，u 和 v 之间不存在链. 因为如果存在链的话，很容易找到 u 和 v 之间的简单链. 但是这也是 G 中的一条简单链，且 G 在 u 和 v 之间只存在一条简单链.

现在，因为 H 在 u 和 v 之间没有链，H 是非连通的. 不难证明(练习 22) H 正好有两个连通分支. 称它们为 H_1 和 H_2. 因为其中每一个都是连通的，且没有回路(为什么)，所以每一个都是树. 另外，每一个的顶点数量都少于 G. 根据归纳假设，如果 n_i 和 e_i 是 H_i 的顶点数量和边的数量，我们有 $n_1=e_1+1$ 及 $n_2=e_2+1$. 现在 $n=n_1+n_2$ 及 $e=e_1+e_2+1$(加上边 $\{u, v\}$). 我们得出结论：

$$n=n_1+n_2=(e_1+1)+(e_2+1)=(e_1+e_2+1)+1=e+1.$$ ■

3.5.4 支撑树[㊁]

假设 $G=(V, E)$ 是一个图，且 $H=(W, F)$ 是一个子图. 我们说 H 是一个**支撑子图**(spanning subgraph)如果 $W=V$. 支撑子图是一棵树时被称为**支撑树**(spanning tree). 例如，对于图 3.76 中的图 G，图 3.76 中所示的 H 和 K 是支撑子图，因为它们的顶点个数与 G 相同. K 是一棵支撑树. 支撑树在组合数学中有着广泛的应用，我们马上将对此进行探讨. 电网的分析可简化成寻找相应图的所有支撑树(参见 Deo[1974]). 通过程序有向图，支撑树也可以用于评估程序运行时间(Deo[1974]，p. 442). 它们也出现于政治学和考古学的排序问题中(Roberts[1979]和 Wilkinson[1971]). 它们也构成网络流中的大量算法和运筹学中求解最小成本问题的基

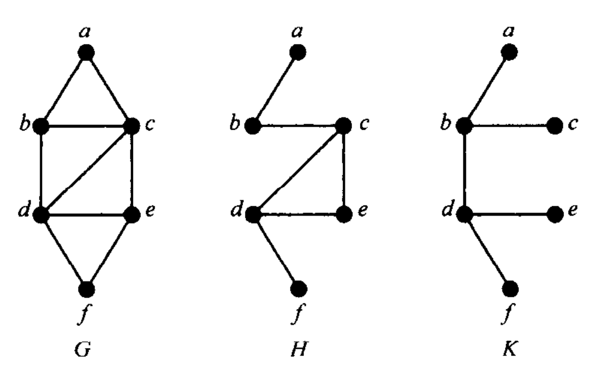


图 3.76 H 和 K 是 G 的支撑子图，K 还是一棵支撑树

㊀ 本小节可以跳过.

㊁ 本小节可以跳过，在 11.1 节或 13.1 节之前学习.

础(第 13 章). Graham 和 Hell[1985]提及了支撑树在计算机设计、通信网络设计、电力网、租线电话网络、接线、运输网络中的链接、流程网络中的管道、网络可靠性、图像处理、自动语音识别、聚类和分类问题等方面的应用. 他们的论文也给出了许多支撑树及其应用的参考文献.

189 现在，我们更详细地给出支撑树的若干应用.

【例 3.28 高速公路的构筑】 假设在一个没有良好高速公路的偏远地区涌现出了几个城市. 理想的是建筑足够多的高速公路，使得或者直接通过高速公路或者通过去其他城市实现各城市之间的往来. 设城市是一个图 G 的顶点，而且由一条边连接每一对城市. 我们希望选取 G 的边的一个子集 F 表示要修建的高速公路，使得如果我们使用 G 的顶点和 F 的边，就有一个连通图. 因此，我们寻找 G 的一个连通支撑子图. 如果我们希望最小化成本，那么我们选择要修建的高速公路一定对应于具有如下性质的 G 的连通支撑子图 H：除去任意一条边将产生一个不再是连通的支撑子图. 但是，如果在 G 的任意连通支撑子图 H 中存在一个回路，那么除去这一回路的任意一条边都不会断开 H(为什么). 因此，H 是连通的且没有回路；即 H 是一棵支撑树. 另外，如果 G 的每一条边上都有表示修建相应的高速公路的成本的权或数字，那么，我们希望寻找一棵在下面的意义下最小的支撑树 F：其边上的权的总和不大于任何其他支撑树的边上的权的总和. 我们将在 13.1 节研究寻找最小支撑树的算法. 有时候，我们也对寻找最大支撑树感兴趣. 寻找它们的过程类似于寻找最小支撑树的过程. 应该注意到的是，如果连接是瓦斯管道、电线连接、铁路、污水管道等等时，也引发类似的最小支撑树的应用. ■

例 3.28 所采用的方法中假设不允许高速公路在所研究城市以外的某一点连接. 在该假设中蕴涵：我们想要或者直接在两个城市之间往来或者通过另一个城市把它们连接起来. 如果我们允许高速公路在所研究城市之外的某一点连接，那么最小化成本时可能实现额外的成本节约. 190 这些成本节约的“额外”点称为*施泰纳点*(Steiner point). 对于施泰纳点的更多信息可以参见 Bern 和 Graham[1989]、Cieslik[1998].

【例 3.29 电话线】 在偏远地区，可以通过公路连接分散的村落，但是没有任何电话服务. 我们希望铺设电话线路，使得每一对村落都可以用电话线连接(不必是直接的). 沿着现存的公路铺设电话线是最便宜的. 沿着哪些公路铺设电话线，可确保每一对村落被连接，且电话线的总长度(这一总长度可能与安装总成本成正比)可以达到最小呢？我们再一次寻找最小支撑树(为什么). ■

【例 3.30 计算机硬件(Ahuja，Magnanti 和 Orlin[1993])】 一台数字计算机有很多由高频线路连接的各种组成部分. 为了减小容量和延迟线路效应，最小化两个部件之间的线路长度是很重要的. 我们需要连通所有组成部分并寻找一棵最小支撑树. ■

【例 3.31 数据存储(Ahuja，Magnanti 和 Orlin[1993]、Kang 等[1977])】 在很多应用问题中，数据被存储在一个二维数组中. 当这一数组的行有很多类似的项，只在少数几个地方不同时，我们可以通过只完全地存储一行，然后再存储其他行与这一参考行(reference row)的差额来节省内存. 构建一个在行 i 和行 j 之间的边上具有权的完全图 G 就是：从第 i 行切换到第 j 行，从第 j 行切换到第 i 行时所需的项改变数量. 我们可以通过选取带有最小数据存储量的那一行作为参考行，然后寻找图 G 的最小支撑树来最小化数据存储. 为什么？ ■

【例 3.32 测量双金属对象的同质性㊀】 最小支撑数已在美国国家标准局以及其他地方都得

㊀ 此内容来自于 Filliben，Kafadar 和 Shier[1983]、Goldman[1981]、Shier[1982].

到应用，如确定双金属对象成分在什么程度上具有同质性. 给定一个这样的对象，通过把原料的一组样品点作为顶点来构建一个图. 对每一个样品点测量其成分并由一条边连接物理上相近的两个样品点. 在边$\{x, y\}$上设置权，该权等于点x与y之间的物理距离乘以0到1之间的同质因数. 如果这两点的成分完全一样则这一因数是0，如果这两点的成分完全相反则等于1，否则就是0～1之间的一个数. 在结果图中寻找最小支撑树. 这棵树的所有权的和是0，表示所有样品点有完全相同的成分. 有一个很高的值则说明原料差异很大. 最小支撑树上边的权的和这一统计量是测量同质性中一个非常有价值的统计量. 根据Goldman[1981]，它可能发展成为一段时期内(相当长)的使用标准. ■ [191]

现在我们给出支撑树的一个主要结果.

定理 3.19 *G是连通的当且仅当它有支撑树.*

证明 假设G是连通的. 那么存在G的一个有最小数量的边的连通支撑子图H. 现在，H没有回路. 因为如果C是H的一个回路，就除去C的任意一条边(不除去相应的顶点)留下G的一个仍连通且边比H少一条的支撑子图. 但是根据H的选择这是不可能的. 因此，H没有回路. 同样根据选择，H是连通的. 因此，H是支撑树.

相反，如果G有一棵支撑树，那么它显然是连通的. ■

可以如下重述定理3.19的证明. 假设我们从连通图G开始. 如果G没有回路，它已经是一棵树. 如果它有一个回路，除去这一回路的任意一条边，留下一个连通图，如果还存在一个回路，除去该回路的任意一条边留下另一个连通图，以此类推.

注意，定理3.19给出确定图G是否连通的方法：简单地测试G是否有一棵支撑树. 实现这一点的算法将在13.1节讨论.

3.5.5 定理3.16的证明和相关结果[⊖]

现在我们给出定理3.16的证明，然后给出一个相关的定理.

定理3.16的证明 在定理3.15中，我们已经证明了该等价的一个方向. 为了证明另一个方向，假设G是连通的，且有$n=e+1$. 根据定理3.19，G有一棵支撑树T. 那么T和G有相同数量的顶点. 根据定理3.15，T有$n-1$条边. 因为G也有$n-1$条边，且因为T的所有边都是G的边，所以有$T=G$. ■

我们还有类似于定理3.16的另一个结果，这一结果将在第13章中讨论.

定理 3.20 *假设G是一个有n个顶点和e条边的图. 那么G是树当且仅当G没有回路且$n=e+1$.*

证明 同上，只剩下证明这一定理的一个方向. 假设G没有回路，且$n=e+1$. 如果G不是连通的，设K_1，K_2，…，K_p是它的连通分支，$p>1$. 设K_i有n_i个顶点. 那么K_i是连通的且没有回路，所以K_i是树. 因此，通过定理3.15，K_i有n_i-1条边. 我们得出结论：G的边数量由下式给出：

$$(n_1-1)+(n_2-1)+\cdots+(n_p-1)=\sum n_i-p=n-p.$$

但是因为$p>1$，$n-p<n-1$，所以有$n\neq e+1$. 这是一个矛盾. ■ [192]

3.5.6 化学键和树的数量

1875年，凯莱(Arthur Cayley)在尝试着枚举出饱和碳氢化合物的异构体，即形如C_kH_{2k+2}的化学混合物时发现了树. 这一工作是化学和生物化学领域的大量图论研究的先驱. 这里，我们

⊖ 本小节可以跳过.

给出凯莱的方法.

在化学中组合数学方法被用来表示分子、群和反应路径.(将在第 9 章中学习的)编码理论是系统化大量可用化学数据的基础,是枚举分子、异构体和有各种性质的族的基础. 图论被用来理解化学混合物、蛋白质等的结构,以及化学结构中的对称性,而且图论对开发化学命名的分类方法也是很有用的. 在计算机辅助分子设计和组合化学的快速发展领域,工业对这些技术的使用不断增加. 对于图论和化学的优秀参考文献,参见 Balaban[1976]、Hansen,Fowler 和 Zheng[2000]、McKee 和 Beineke[1997]、Rouvray 和 Balaban[1979]. 也可参见《化学教育杂志》1992 年 9 月刊,其中特载了化学中的图论. 我们将在 6.4 节和第 8 章给出组合数学对化学的其他应用.

异构体 C_kH_{2k+2} 可以通过用字母 C 代表碳原子并用字母 H 代表氢原子来表示,如果两个原子在给定的混合物中是键合的,那么连接这两个原子. 例如,图 3.77 给出了甲烷和乙烷的表示. 我们可以使用由顶点取代字母的图取代这些图,如图 3.78 所示. 我们称这些图为**键合图**(bond graph). 注意,给定一个饱和碳氢化合物的键合图,可以用明确的方式使用字母 C 和 H 重新给顶点赋标签:如果一个顶点与其他 4 个顶点键合,则给其赋标签 C(碳的化合价为 4),如果它与另一个顶点键合则赋标签 H(H 的化合价是 1). 键合图的每一个顶点或者与一个或者与四个顶点键合. 某些其他的饱和碳氢化合物键合图如图 3.79 所示.

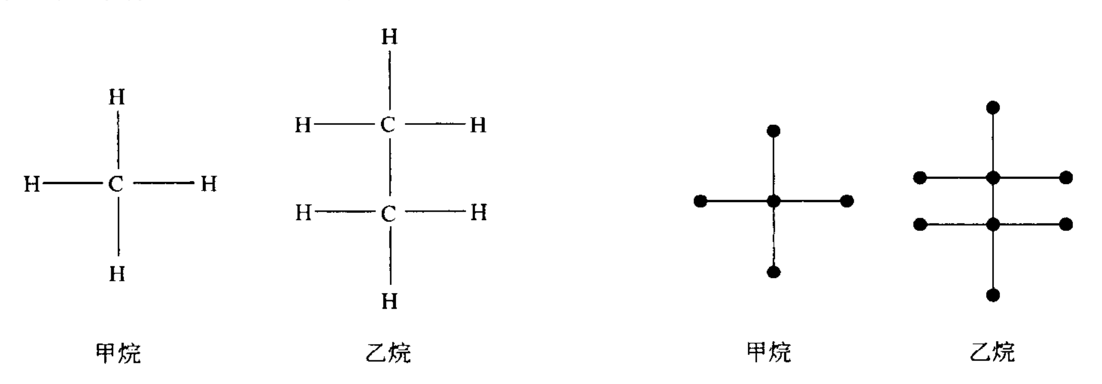


图 3.77 两个饱和碳氢化合物　　图 3.78 图 3.77 中的饱和碳氢化合物的键合图

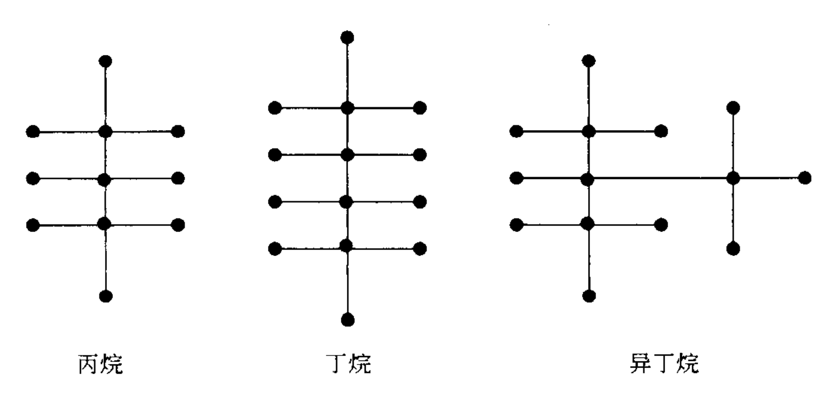


图 3.79 更多键合图

图 3.78 和图 3.79 中的图都是树. 我们将证明这不是偶然的,即每一个饱和碳氢化合物的键合图都是树.

回想一下,图中的顶点的度是它的邻居数量,且根据定理 3.1,顶点的度的和是边数量的两倍. 现在在 C_kH_{2k+2} 的键合图中,有

$$k + 2k + 2 = 3k + 2$$

个顶点. 另外，每一个碳顶点有度4，而每一个氢顶点有度1. 因此，度的和是

$$4k+1(2k+2)=6k+2.$$

193～194

边的数量是这个数的一半，即 $3k+1$. 因此，边的数量恰好比顶点的数量少1. 因为化合物 C_kH_{2k+2} 的键合图一定是连通的，所以根据定理3.16，这个键合图是树.

现在，凯莱把枚举所有可能的饱和碳氢化合物的问题，抽象成枚举每一个顶点有度1或4的所有树的问题. 他发现从枚举所有树开始更加容易. 在这一过程中，他从理论上发现了之前不知道的一些饱和碳氢化合物的键合图，而且预言了它们的存在. 这些饱和碳氢化合物后来都被发现了.

显然，是否认为特定类型的图已被标记(参见3.1.3节的讨论)，在计数该类型的不同图的数量上会产生很大的不同. 特别地，凯莱发现，对于 $n\geq 2$，存在 n^{n-2} 个有 n 个顶点的标签树. 在这里，我们将给出这一结果的一个证明. 对于其他证明的概览，可以参见Moon[1967](还可参见Harary和Palmer[1973]、Shor[1995]、Takács[1990]).

定理3.21(Cayley[1889]) *如果 $n\geq 2$，那么存在 n^{n-2} 棵不同的有 n 个顶点的标签树.*

设 $N(d_1, d_2, \cdots, d_n)$ 计数有 n 个顶点的标签树的数量，且顶点标签为 i，有度 d_i+1. 我们首先注意到下面的结果，这将在练习34中证明.

定理3.22 *如果 $n\geq 2$ 且 d_i 是非负整数，那么*

$$N(d_1,d_2,\cdots,d_n)=\begin{cases}0 & \text{如果}\sum_{i=1}^{n}d_i\neq n-2\\ C(n-2;d_1,d_2,\cdots,d_n) & \text{如果}\sum_{i=1}^{n}d_i=n-2.\end{cases}\tag{3.21}$$

在这一定理中，

$$C(n-2;d_1,d_2,\cdots,d_n)=\frac{(n-2)!}{d_1!d_2!\cdots d_n!}$$

是2.11节所学过的多项式系数.

为了说明定理3.22，注意

$$N(1,3,0,0,0,0)=C(4;1,3,0,0,0,0)=\frac{4!}{1!3!0!0!0!0!}=4.$$

有4棵6个顶点的标签树且顶点1有度2，顶点2有度4，剩余的顶点有度1. 图3.80给出这四棵树.

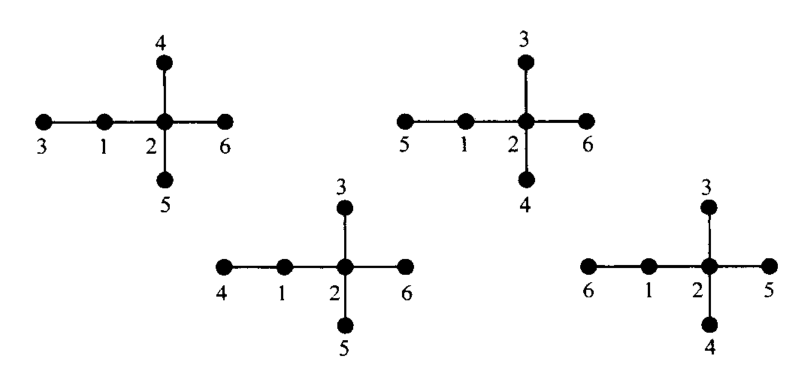


图3.80 分别带有度1+1、3+1、0+1、0+1、0+1、0+1的四棵标签树

现在，凯莱定理可以作为定理3.22的一个推论. n 个顶点的标签树的数量 $T(n)$ 由下式给出：

$$T(n)=\sum\left\{N(d_1,d_2,\cdots,d_n):d_i\geqslant 0,\sum_{i=1}^{n}d_i=n-2\right\},$$

这等同于

195

$$T(n)=\sum\left\{C(n-2;d_1,d_2,\cdots,d_n):d_i\geqslant 0,\sum_{i=1}^{n}d_i=n-2\right\}.$$

现在，很容易看到二项式展开式(定理 2.7)扩展成下面的*多项式展开式*：如果 p 是一个正整数，那么

$$(a_1+a_2+\cdots+a_k)^p=\sum\left\{C(p;d_1,d_2,\cdots,d_k)a_1^{d_1}a_2^{d_2}\cdots a_k^{d_k}:d_i\geqslant 0,\sum_{i=1}^{k}d_i=p\right\}.$$

因此，取 $k=n$，$p=n-2$，$a_1=a_2=\cdots=a_k=1$，我们有

$$T(n)=(1+1+\cdots+1)^{n-2}=n^{n-2}.$$

这就是凯莱定理.

3.5.7　种系发生树的重建

在例 3.27 中，我们引入了进化树，或种系发生树的概念. 更一般的种系发生树在它的边上有代表两个进化事件之间的时间历程的实数权. 生物学(和其他领域，如语言学)中的一个重要问题是，根据有关现存物种的某些信息重新构建种系发生树. 从生物学观点对这一课题的一般性介绍可以在 Fitch[1999]中找到，从组合数学的观点对此的介绍可以在 Gusfield[1997]中找到. 种系发生树重建的一般方法开始于现存物种的 DNA 序列. *基于序列的方法*(sequence-based method)直接使用这些序列重建种系发生树. 相反，*基于距离的方法*(distance-based method)首先计算所有序列对的距离，然后使用这些距离重建种系发生树. 更详细地，它们开始于 n 个物种和一个 $n\times n$ 的对称矩阵 D，其 i，j 项 $D(i, j)$给出这两个物种 i 和 j 之间的计算距离. 然后，它们寻找一棵树，在这棵树中，给定的物种是叶子(leaf)，即只带有一个邻居的顶点，且 $D(i, j)$等于或非常近接于这棵树中的 i 和 j 之间的距离 $d(i, j)$. 这里，我们通过计算这棵树上唯一的从 i 到 j 的简单链上的权的和来度量具有边权的树中顶点 i 和 j 之间的距离 $d(i, j)$. (定理 3.18 说，在任意一棵树上，在给定的两个顶点之间总是存在唯一的简单链.)

196

设 D 是一个对角线上的元素为 0 的 $n\times n$ 对称矩阵. 我们说 D 是一个*附加距离矩阵*(additive distance matrix)，如果我们能够找到有 n 个叶的边加权树，其中 n 对应于给定的物种，使得 $d(i, j)=D(i, j)$. 例如，图 3.81 的矩阵 D 是一个附加距离矩阵，且该图中的 T 是这个矩阵所对应的加权树. 例如，在这棵树中，从 1 到 4 的唯一简单链上的权是 2，3 和 1，所以 $d(1, 4)=6$，这对应于 $D(1, 4)$. 什么样的矩阵是附加距离矩阵呢?

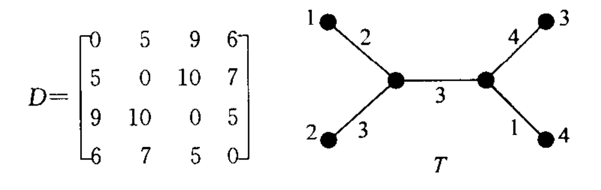


图 3.81　一个附加距离矩阵 D 和对应的加权树

定理 3.23(Buneman[1971])　*对角线上元素是 0 的对称矩阵 D 是附加距离矩阵，当且仅当对于所有的 i，j，k，l，三对和 $D(i, j)+D(k, l)$，$D(i, k)+D(j, l)$，$D(i, l)+D(j, k)$中，最大的两个相等.*

为了说明这一定理，我们注意到在图 3.81 中，$D(1, 2)+D(3, 4)=9$，$D(1, 3)+D(2, 4)=15$，$D(1, 4)+D(2, 3)=15$. 在可能的条件下，存在重建正好对应于附加距离矩阵的边加权树的好算法，事实上是 $O(n^2)$算法. 对于其他参考文献，可以参见 Gusfield[1997].

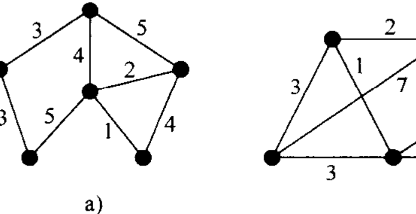
如果一个矩阵不是附加距离矩阵，那么我们可能希望寻找一个"最接近"的边加权树. 使用各种方法可以使这一问题更精确，但是几乎所有方法都已证明很难得到准确解(更准确地说，

这一问题是NP难题). 一个这样的问题是寻找一个附加距离矩阵D'使得$\max_{i,j}|D(i,j)-D'(i,j)|$尽可能小(该问题是NP难题). Agarwala等[1996]找到一个近似最优解决方案的算法，即寻找一个边加权树且保证$\max_{i,j}|D(i,j)-D'(i,j)|$至多是最优的三倍. Cohen和Farach[1997]改进了这一算法. 基于Buneman定理(定理3.23)的其他算法在Erdös等[1997, 1999]中得到描述.

197

本节练习

1. 如果"ZIP代码"是由四个数字组成的，且每一个都是0、1或者2. 画出分类邮件的分类树.
2. 寻找四个顶点的所有非同构树.
3. 寻找五个顶点的所有非同构树.
4. 确定：
   (a)有10条边的树中的顶点个数.
   (b)有10个顶点的树中的边的个数.
5. 对于图3.76中的各图，核实定理3.1.
6. 给出满足$n=e+1$但不是一棵树的图G的例子.
7. 对于图3.23中的每一个图，或者寻找一棵支撑树，或者证明不存在支撑树.
8. 在图3.82所示的边上带权的每个图中，寻找带有最小总权的支撑树.
9. 森林(forest)是一个图，它的每一个连通分支都是一棵树. 如果一个森林有n个顶点和k个分支，它有多少条边?
10. 定理3.18的唯一性的一个简单"证明"如下所示. 假设C_1和C_2是连接x和y的两个不同的简单链. 那么C_1后面跟着的C_2是一个闭链. 但是，如果一个图有闭链，那么它必定有回路. 证明后面的陈述为假.



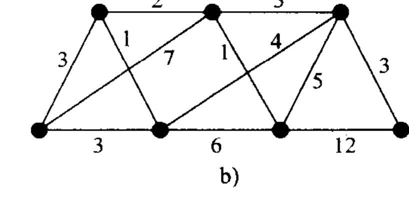


图3.82 边上带权的图

11. 在一个有15条边的连通图中，顶点的最大可能数量是多少?
12. 在一个有25个顶点的连通图中，边的最小可能数量是多少?
13. 在一个有15条边和三个分支的图中，顶点的最大数量是多少?
14. 证明定理3.18的逆定理；即如果G是任意图，且任意两个顶点都由唯一的简单链连接，那么G是树.

198

15. 证明：如果用一条边连接一棵树的两个非邻接顶点，那么这一结果图有一条回路.
16. 证明：如果从一棵树中除去任意一条边，那么结果图是非连通的.
17. 如果我们有一个(连通)电网，它有e个元件(边)和n个节点(顶点)，为了清除这一网络中的所有回路，我们必须消除的元素的最小数量是多少?
18. 如果G是一棵有n个顶点的树，证明它的色多项式由下式给出：
$$P(G,x)=x(x-1)^{n-1}.$$
19. 使用练习18的结果确定一棵树的色数.
20. 通过证明每一棵树是二部的来确定树的色数.
21. 证明练习18的结果的逆命题为真，即如果
$$P(G,x)=x(x-1)^{n-1},$$
那么G是一棵树.
22. 假设G是一棵树，$\{u, v\}$是G的一条边，H是通过从G消除边$\{u, v\}$得到的，但是保留顶点u和v. 证明H正好有两个连通分支.(不能使用本节中除定理3.18外的任何定理.)

23. (Peschon 和 Ross[1982])在一个配电系统中，某些配置是由连通的电线连接起来的．使用一个开关系统打开或关闭这些线路．打开的线路的全体必须有两个性质：(1)每一个配置必须在一条打开的线路上，(2)没有打开线路的回路，因为打开回路中一条打开线路的短路将关闭这一回路中的所有线路．讨论寻找需要打开的开关中的数学问题．
24. (Ahuja，Magnanti 和 Orlin[1993]、Prim[1957])一个情报机关的间谍每一位都知道如何互相联系．然而，在两个间谍 i 和 j 之间传送的信息被拦截的概率是 p_{ij}．如果我们想要确保所有间谍得到一个信息，且最小化信息被拦截的概率，哪些间谍应该把这一信息传给哪些间谍？作为支撑树问题来阐述这一问题．[*提示*：你将需要使用对数．]
25. 这里是寻找一个连通图 G 的支撑树的算法．任意挑选出一个顶点并标记它．包括连接没有被标记的邻接顶点的边并标记那个邻接顶点．在每一步，都从最后被标记的顶点到一个没有被标记的顶点不断增加一条边，直到没有办法继续为止．然后返回到最近被标记的顶点，并从这一顶点继续下去．当所有顶点都已被标记时停止．(这一过程称为*深度优先搜索*(depth first search)，在 11.1 节我们将讨论这一课题．)
   (a)使用图 3.82 中的图说明这一算法(不考虑权)．
   (b)证明：存在使用此方法无法找到的一个图的一棵支撑树．
26. 假设一个化合物 C_kH_m 有一个键合图，这个键合图连通且没有回路．证明 m 必定是 $2k+2$．

199

27. 确定 K_n 的支撑树的数量．
28. 通过寻找下列情况的所有标签树核实凯莱定理．
   (a)有三个顶点．
   (b)有四个顶点．
29. 存在符合下列各条件的七顶点树吗：
   (a)每个顶点有度 1?
   (b)两个顶点有度 1 且五个顶点有度 2?
   (c)五个顶点有度 1 且两个顶点有度 2?
   (d)各顶点有度 2，2，2，3，1，1，1?
30. 存在有五个顶点且其中两个顶点度为 3 的树吗?
31. 在下列各情况中，通过公式确定满足给定度的条件的标签树的数量，并画出所讨论的树．
   (a)顶点 1，2 和 3 有度 2，顶点 4 和 5 有度 1．
   (b)顶点 1 有度 2，顶点 2 有度 3，顶点 3，4 和 5 有度 1．
   (c)顶点 1 有度 3，顶点 2 和 3 有度 2，顶点 4 和 5 和 6 有度 1．
32. 寻找下列情况的标签树的数量：
   (a)六个顶点，其中四个有度 2．
   (b)八个顶点，其中六个有度 2．
   (c)五个顶点，其中正好有三个有度 1．
   (d)六个顶点，其中正好有三个有度 1．
33. 证明：如果 $\sum_{i=1}^{n} d_i \neq n-2$，则 $N(d_1, d_2, \cdots, d_n)=0$．[*提示*：对边数计数．]
34. 这一练习给出定理 3.22 的证明框架．使用等式(3.21)的右边定义 $M(d_1, d_2, \cdots, d_n)$．只要证明如果 $n\geqslant 2$，所有 d_i 是非负的且 $\sum_{i=1}^{n} d_i = n-2$，那么

$$N(d_1, d_2, \cdots, d_n) = M(d_1, d_2, \cdots, d_n) \tag{3.22}$$

   即可．
   (a)在给定假设下，对 $n=2$ 证实(3.22)式．
   (b)在给定假设下，证明：对于某个 i 有 $d_i=0$．

(c)假设(b)中的 i 是 n. 证明下面的等式：

$$N(d_1,d_2,\cdots,d_{n-1},0)=N(d_1-1,d_2,d_3,\cdots,d_{n-1}) + N(d_1,d_2-1,d_3,\cdots,d_{n-1})+\cdots + N(d_1,d_2,d_3,\cdots,d_{n-2},d_{n-1}-1), \quad (3.23)$$

其中当且仅当 $d_k>0$ 时项 $N(d_1,\ d_2,\ \cdots,\ d_{k-1},\ d_k-1,\ d_{k+1},\ \cdots,\ d_{n-1})$ 出现在(3.23)式的右边.

(d)证明 M 也满足(3.23)式.

(e)通过对 n 作归纳证明(3.22)式(用第6章的术语来说，这一证明本质上相当于证明：如果 M 和 N 满足相同的递归式和相同的初始条件，那么有 $M=N$).

200

35. 确定下列矩阵哪些是附加距离矩阵.

(a) $\begin{bmatrix}0&1&3\\1&0&2\\3&2&0\end{bmatrix}$ (b) $\begin{bmatrix}0&1&4\\1&0&2\\4&2&0\end{bmatrix}$ (c) $\begin{bmatrix}0&1&3&9\\1&0&2&2\\3&2&0&1\\9&2&1&0\end{bmatrix}$.

36. (a)证明在图3.83的树a中，定理3.23中的所有三个和都相等.

(b)证明对于图3.83的树b上述结论为假.

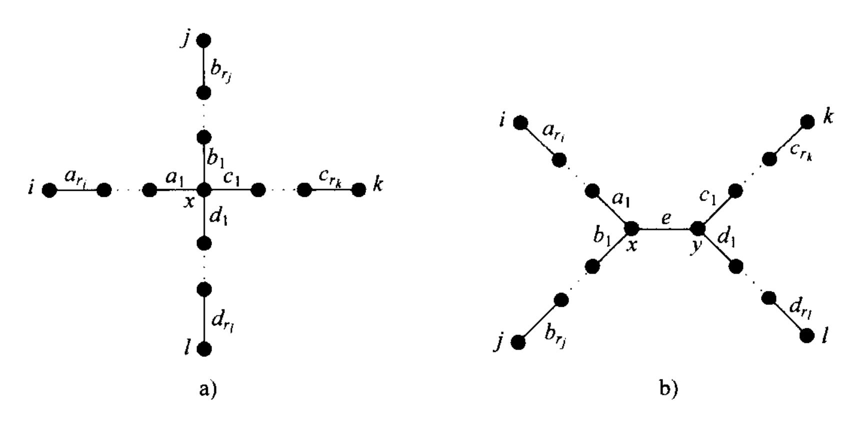


图3.83 边加权树. 在这些树中，a中从 x 到 i、到 j、到 k、到 l 的链的长度和b中从 x 到 i、到 j 和 y 到 k、到 l 的链的长度是任意的. 而且，可以把任意正权 a_i、b_i、c_i、d_i、e 分配给这些边

37. 假设 T 是一棵边加权树且有

$$d(i,j)+d(k,l)\leqslant d(i,k)+d(j,l)=d(i,l)+d(j,k).$$

证明顶点 i，j，k，l 被定位于图3.83的a或者b中.

38. 假设 T 是一棵种系发生树，且每一个顶点至多有三个邻居. 当每一个"进化事件"涉及一个种群分离成两个新种群时出现这一情况. 根据练习37证明对于所有的 i，j，k，l，定理3.23中的三个和之一严格小于其他两个和.

201

3.6 根树的应用[⊖]

3.6.1 定义

使用树搜索一个表格或文件是计算机科学中最重要的操作之一. 在例2.18中，我们讨论了搜索一个文件寻找某个人的键(识别数字)的问题，并指出存在更有效的搜索方法而不是只简单地从列表的头走到尾. 本节给出如何使用搜索树做到这一点. 然后，对于计算机科学中的另一个重要问题我们给出如何使用树：把一组给定元素按其自然顺序排序，如给定的数字列表. 最后，

⊖ 本节可以跳过.

我们使用关键字的思想阐述 3.5.7 节中讨论的种系发生树重建问题的一个不同方法.

在定义搜索树之前，我们注意到图 3.72 到图 3.74 中的每一棵树都是按下面的方式画出来的. 每一个顶点有*层次*(level)0，1，2，…，k. 正好存在一个层次是 0 的顶点，*根*(root). 所有邻接顶点正好相差一个层次，且位于层次 $i+1$ 上的每一个顶点正好与位于层次 i 上的一个顶点邻接. 这样的树称为*根树*(rooted tree)(不难证明，在指定一个顶点作为根之后，每一棵树都可被认为是根树；参见练习 5). 数字 k 称为这一根树的*高度*(height). 在图 3.72 中的例子中，层次 0(根)的顶点是 Boss；层次 1 的顶点是 Jones、Smith 和 Brown；层次 2 的顶点是 Johnson、Jackson 等等；层次 3 的顶点是 Allen 和 Baker 等等. 高度是 3. 在一棵根树中，所有与顶点 u 邻接且位于比 u 低的层次上的顶点称为 u 的*孩子*(children). 例如，在图 3.72 中，Brown 的孩子是 White、Black 和 Green. 在一棵树上，通过一条顶点链与 u 相连且位于比 u 低的层次上的所有顶点称为 u 的*后代*(descendant). 因此，在我们的例子中，Brown 的后代是 White、Black、Green、Engel、Freyer 和 Gold.

一棵根树称为 *m 元*(m-ary)树，如果每一个顶点有 m 个或少于 m 个孩子. 2 元根树称为*二叉树*(binary tree). 一棵根树称为*完全 m 元*(complete m-ary)树，如果每一个顶点或有 0 个或 m 个孩子. 图 3.84 给出一个完全二叉树. 在一棵二叉树中，我们将假设一个顶点的任意孩子被指明为*左孩子*(left child)或*右孩子*(right child).

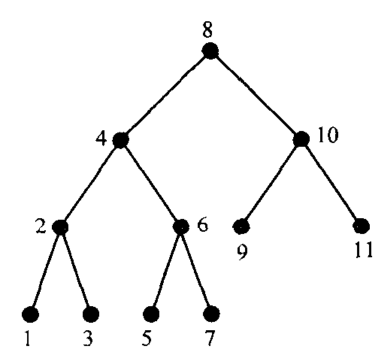


图 3.84　一个完全二叉搜索树

【例 3.33 代码树】 当需要传输数据时，每一个符号被编码，或被指定一个二进制串. 假设数据由成绩组成：15 个 A、27 个 B、13 个 C、4 个 D 和 1 个 F. 如果每一个位串的长度相同，那么我们需要长度为三(或更长)的位串表示这五个符号. 因此这一数据集合至少需要

202

$$(15+27+13+4+1)\cdot 3=180\text{ 位.}$$

如果我们是要最小化需要传输的位数，那么将不能使用长度为 4 或更长的位串.

如果允许我们使用可变长度的位串，那么就能节省更多. 假设我们如下表示成绩：A：01，B：1，C：111，D：10，F：010. 现在我们只需要

$$(15\cdot 2)+(27\cdot 1)+(13\cdot 3)+(4\cdot 2)+(1\cdot 3)=107\text{ 位.}$$

尽管这节省了传输的位数，但对于这一编码还存在一个问题. 考虑下面的传输：01011…. 当接收者开始解码时，第一个成绩是什么？可能是一个 A，因为传输开始于 01. 然而，也可以解释成一个 F，因为传输也开始于 010. 这一问题的出现是因为一个位串是某个其他位串的前缀. 我们需要寻找一种这一情况不会发生的编码. 这样的编码称为*前缀码*(prefix code).

容易找到前缀码(事实上，上面的长度为 3 的位串编码就是前缀码). 考虑图 3.85 中的二叉树. 在该树中，左分支被标记为 0，右分支被标记为 1. 通过查看从根到这个顶点的唯一链，可生成一个与每个顶点相关联的位串. 表示各成绩的顶点相应地被标记. 注意到标签为 B(010)的顶点的唯一链开始于根且走向左侧(0)，然后到右侧(1)再到左侧(0).

这一前缀码在把长度为 3 的位串指定给每一个成绩的编码时可以节省位数. 但是，这是我们能做的最好的吗？同上面一样，与前缀码相关的树称为*代码树*(code tree). 图 3.85 仅是关于该例的代码树的一个例子. 最小化传输数据位数的问题等价于寻找一棵最优的代码树. 给定一

个符号集合和这些符号的使用频率，Huffman[1952]给出构建最优代码树的算法．下面算法输出的最优代码树有时称为赫夫曼树(Huffman tree)． 203

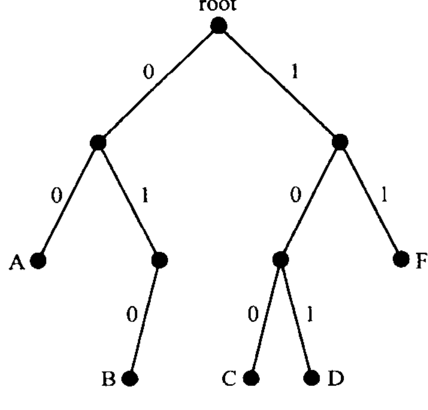


图 3.85 一棵代码树

算法 3.2：最优代码树

输入： n 个独立顶点(根树)的森林 S，带有标签和权，标签是符号$\{s_1, s_2, \cdots, s_n\}$，权是频率$\{f_1, f_2, \cdots, f_n\}$．

输出： 一棵最优树．

步骤 1. 在 S 中寻找两棵有最小根权 f_i 和 f_j 的树 T 和 T'．

步骤 2. 创建一棵新的有根 s_{ij} 和权 $f_{ij}=f_i+f_j$ 并且以 T 和 T' 作为它的孩子的完全二叉树 T''．

步骤 3. 用 T'' 取代 S 中的 T 和 T'．如果 S 是一棵树，停止并输出 S，否则，返回到步骤 1．

作为一个例子，假设需要传输下面的数据：15 个 A，4 个 B，10 个 C，15 个 D，11 个及格和 3 个不及格．图 3.86 给出使用算法 3.2 的一个最优代码树的构建过程．注意，如果所有位串的长度必须相同，那么 6 个不同的信息(A，B，C，D，及格，不及格)至少需要长度为 3 的位串．因此，传输的位的总数量等于$(15+4+10+15+11+3)\cdot 3=174$．使用算法 3.2，我们发现只需要

$$(15\cdot 2)+(4\cdot 4)+(10\cdot 3)+(15\cdot 2)+(11\cdot 2)+(3\cdot 4)=140$$

个位，节约了大约 20%．我们在练习中证明这一算法的最优性(参见练习 20)．第 10 章有更多关于代码的内容． 204

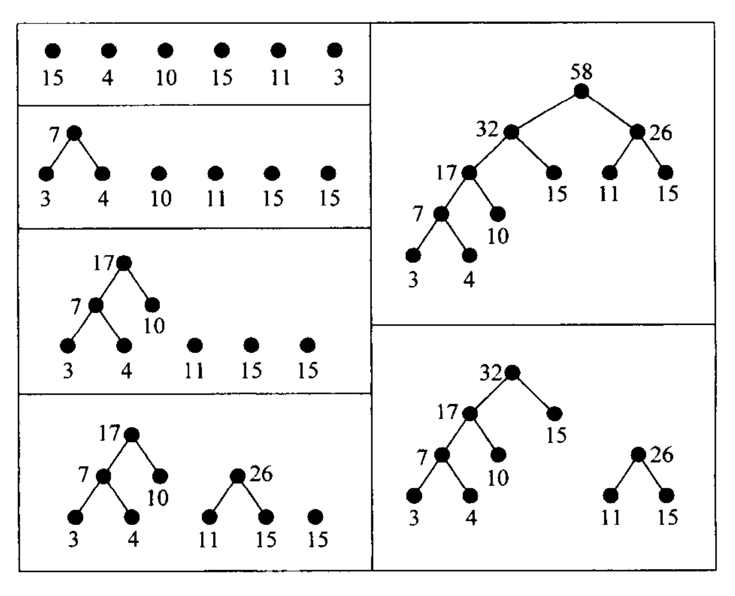


图 3.86 使用算法 3.2 传输 15 个 A、4 个 B、10 个 C、15 个 D、11 个及格和 3 个不及格的最优代码树(图/算法开始于左上角，且沿逆时针进行) ■

3.6.2 搜索树

假设我们有一个 n 个名字的文件．设 T 是任意二叉(根)树．按下面的方法使用一个键(一个实数)为 T 的每一个顶点 u 赋标签：如果 u 有一个孩子，那么它的左孩子和左孩子的所有后代得到比 u 的键的实数小的键，它的右孩子和右孩子的所有后代得到比 u 的键的实数大的键．带有这

样标签的二叉树称为二叉搜索树(binary search tree)(可以为每一个二叉树找到这样的标签；参见下面的内容)．图 3.84 给出一个完全二叉搜索树，在这棵二叉树上，键是整数 1，2，…，11．现在给定一个键 k，为了定位这个键，我们从根开始搜索这棵二叉树．在每个阶段，我们寻找被检查的顶点 x 的键．开始时 x 是根．我们看 k 是等于、小于，还是大于 x 的键．如果等于，那么我们就已找到了所要的文件．如果小于，我们接下来看 x 的左孩子；如果大于，我们接下来看 x 的右孩子．持续同样的过程．例如，使用图 3.84 的二叉搜索树，为了寻找键 7，我们从键 8 开始，这是树的根．然后我们走到键 4(因为 7 小于 8)，然后走到键 6(因为 7 大于 4)，然后走到键 7(因为 7 大于 6)．注意，我们的搜索花了四步而不是按照列表 1，2，…，11 的顺序搜索所需的七步．关于二叉搜索树的其他应用参见练习 23．

205

假设我们能够找到一个二叉搜索树．最坏情况下文件搜索的计算复杂度是什么呢？这个复杂度是从根降到没有孩子的顶点的最长链中顶点的个数；即它比二叉搜索树的高度大 1．显然，如果我们找到高度最小的二叉搜索树，那么可以最小化这个计算复杂度．现在，任意的二叉树都可以改成二叉搜索树．关于其证明和算法不妨参见 Reingold，Nievergelt 和 Deo[1977]．因此，留给我们下面的问题．给定 n，有 n 个顶点且高度为 h 的二叉树的最小 h 是什么？我们将通过下面的定理回答这一问题，这一定理的证明已在 3.6.3 节中给出．其中，$\lceil a \rceil$是比 a 大或等于 a 的最小整数．

定理 3.24　*n 个顶点的二叉树的最小高度是 $\lceil \log_2(n+1) \rceil - 1$．*

图 3.84 中的二叉树是有 11 个顶点和最小高度 3 的二叉树，因为

$$\lceil \log_2(n+1) \rceil = \lceil \log_2 12 \rceil = \lceil 3.58 \rceil = 4.$$

总之，定理 3.24 给出使用二叉搜索树搜索文件的计算复杂度的一个对数下界：

$$\lceil \log_2(n+1) \rceil - 1 + 1 = \lceil \log_2(n+1) \rceil.$$

当然，可以通过寻找最小高度二叉搜索树而得到这一下界，这样的二叉搜索树总是可以找到的(参见定理 3.24 的证明)．这一结果概括如下．

推论 3.24.1　*使用二叉搜索树的文件搜索的计算复杂度是 $\lceil \log_2(n+1) \rceil$．*

一般来说，这一对数复杂度比我们在例 2.18 中得到的通过按顺序查看列表中的项来搜索文件的复杂度 n 要好得多．因为当 n 增大时，对数 $\lceil \log_2(n+1) \rceil$ 会比 n 小的多．

3.6.3　定理 3.24 的证明[⊖]

为了证明定理 3.24，我们首先证明下面的定理．

定理 3.25　*如果 T 是一个有 n 个顶点且高度为 h 的二叉树，那么 $n \leqslant 2^{h+1} - 1$．*

证明　层次 0 上存在一个顶点，层次 1 上至多有 $2^1=2$ 个顶点，层次 2 上至多有 $2^2=4$ 个顶点，层次 3 上至多有 $2^3=8$ 个顶点，…，层次 h 上至多有 2^h 个顶点．因此，有

206

$$n \leqslant 1 + 2^1 + 2^2 + 2^3 + \cdots + 2^h. \tag{3.24}$$

现在，我们使用一般公式

$$1 + x + x^2 + \cdots + x^h = \frac{1 - x^{h+1}}{1 - x}, \quad x \neq 1, \tag{3.25}$$

这一公式是本书中一个非常有用的公式．把 $x=2$ 代入(3.25)式中并使用(3.24)式，我们得到

$$n \leqslant \frac{1 - 2^{h+1}}{1 - 2} = 2^{h+1} - 1.$$

■

⊖　本小节可以跳过．

定理 3.24 的证明 我们有

$$2^{h+1} \geqslant n+1,$$
$$h+1 \geqslant \log_2(n+1),$$
$$h \geqslant \log_2(n+1)-1.$$

因为 h 是一个整数，所以

$$h \geqslant \lceil \log_2(n+1) \rceil - 1.$$

所以，每一个有 n 个顶点的二叉树的高度至少是 $\lceil \log_2(n+1) \rceil - 1$. 显然，可以直接证明总是存在有 n 个顶点的二叉树，其高度正好等于 $p=\lceil \log_2(n+1) \rceil - 1$. 事实上，没有孩子的顶点都在层次 p 或 $p-1$ 上的任意完全二叉树就是这样的二叉树[⊖]. ■

3.6.4 排序

计算机科学中的一个基础问题是以自然顺序放置一组项的问题，这通常是根据某个数值. 我们称这样的问题为*排序问题*(sorting problem). 我们研究通过项对的比较进行排序的问题.

通过比较进行排序的任意算法可以由一棵称为*决策树*(decision tree)的(完全)二叉树表示. 图 3.87给出排序三个数 a 、b 和 c 的计算机程序的决策树. 在每一种情况中，树的顶点对应于一个检测问题或一个输出. 在每一个检测问题顶点处，如果检测问题的答案为"是"则控制转向左孩子，否则控制转向右孩子. 输出顶点由方格表示，检测顶点由椭圆表示. 由图 3.87的决策树所示的这一算法的复杂度，是在最坏情况下达到一个决策所需的步骤数(比较). 因为输出对应于没有孩子的顶点，所以通过寻找从根到 T 的没有孩子的顶点的最长链的顶点个数，再减去 1，即通过确定二叉树 T 的高度可以得到这一复杂度. 在我们的例子中，这一高度是 3.

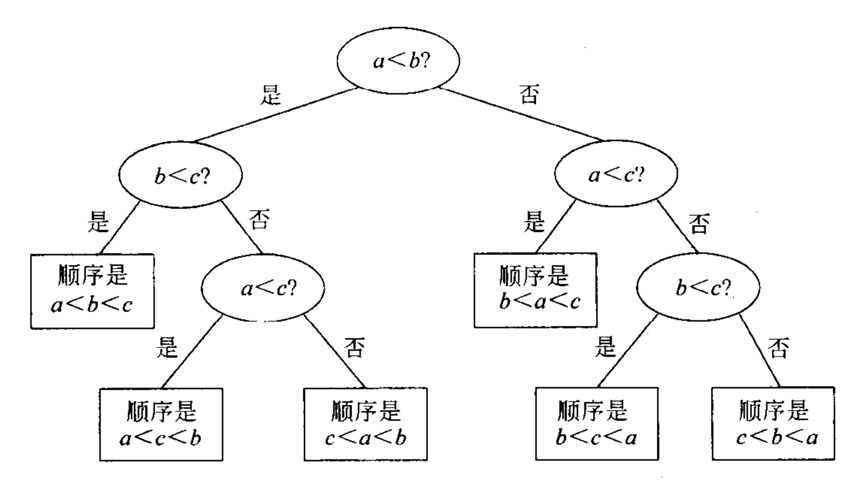


图 3.87 关于数的集合$\{a, b, c\}$排序的一棵决策树 T

在一棵根树中，我们称没有孩子的顶点为*叶子*(对应于叶子的内容已在 3.5.7 节中定义). 注意，为了给一组 p 个不同项目排序，有 $p!$ 种可能的顺序，所以排序的决策树将至少有 $p!$ 个叶子(我们说"至少"是因为若干从一个根出发的链可能有相同的顺序). 现在，我们有下面的定理，其证明留作练习(练习 33).

定理 3.26 *高度为 h 的二叉树至多有 2^h 个叶子.*

⊖ 这样的二叉树称为是*平衡的*(balanced). 图 3.84 是一个平衡二叉树的例子.

通过图 3.74 或图 3.84 的二叉树很容易说明这一定理.

定理 3.27　通过逐对比较排序 $p\geqslant4$ 个项的任意算法在最坏的情况下至少需要 $cp\log_2 p!$ 次比较，其中 c 是一个正常量.

证明　我们已经观察到对 p 个项排序的任意决策树必须至少有 $p!$ 个叶子. 因此，在最坏的情况下的比较次数即树 T 的高度，必须至少是 $\log_2 p!$（因为 $2^{\log_2 p!}=p!$）. 现在，对于 $p\geqslant1$，有

$$p!\geqslant p(p-1)(p-2)\cdots\left(\left\lceil\frac{p}{2}\right\rceil\right)\geqslant\left(\frac{p}{2}\right)^{p/2}.$$

207～208

因此，对于 $p\geqslant4$，有

$$\log_2 p!\geqslant\log_2\left(\frac{p}{2}\right)^{p/2}=\frac{p}{2}\log_2\left(\frac{p}{2}\right)\geqslant\frac{p}{4}\log_2 p=\frac{1}{4}p\log_2 p.$$ ■

有很多确实达到定理 3.27 的下界的排序算法，即可以通过 $p\log_2 p$ 的常量倍步骤实现排序. 其中已知较好算法是堆排序. 在下面和练习中我们讨论两个著名的排序算法——冒泡排序和快速排序，这两个排序达不到定理 3.27 中的下界. 关于所有三种算法的更仔细讨论，不妨参见 Baase[1992]、Brassard 和 Bratley[1995]、Manber[1989]. 注意 $cp\log_2 p\leqslant cp^2$，所以取 $cp\log_2 p$ 步的算法一定是多项式界定的算法.

在称为冒泡排序(bubble sort)的算法中，我们开始于一个有 p 个(不同)项的有序集合. 希望把这些项放在它们的正确(递增)顺序上. 我们不断地比较列表中第 i 项和第$(i+1)$项，如果第 i 项大于第$(i+1)$项，那么交换它们. 这一算法之所以被称为冒泡排序，是因为较大的项总是上升到顶层，就像香槟酒里的泡泡一样. 下面，更形式地描述这一算法.

算法 3.3：冒泡排序

输入：p 个项的有序列表 a_1，a_2，…，a_p.

输出：按递增顺序的 a_1，a_2，…，a_p 的列表.

步骤 1. 令 $m=p-1$.

步骤 2. 对于 $i=1$，2，…，m，如果 $a_i>a_{i+1}$，则交换 a_i 和 a_{i+1}.

步骤 3. 把 m 减 1. 如果 m 现在是 0，停止并输出这一顺序. 否则返回步骤 2.

为了说明冒泡排序，假设我们从顺序 516423 开始. 首先我们令 $m=p-1=5$. 比较 5 和 1 并交换它们，得到 156423. 然后比较 5 和 6，保留这一顺序. 比较 6 和 4 并交换它们，得到 154623. 接下，比较 6 和 2 并交换它们，得到 154263. 最后，比较 6 和 3 并交换它们，得到 154236. 我们把 m 递减到 4，并重复这一过程，相继得到 154236、145236、142536 和 142356. 注意我们不必比较 5 和 6，因为 m 现在是 4. 我们把 m 递减到 3 并重复这一过程，得到 142356、124356 和 123456. 于是设 m 等于 2，我们得到 123456 和 123456. 注意，没有再需要交换的. 接下来，我们设 $m=1$. 没有需要交换的. 最后我们设 $m=0$，并输出这一顺序 123456.

关于顺序 a，b，c，d 的冒泡排序的部分决策树如图 3.88 所示.

注意冒泡排序需要 $p(p-1)/2$ 次比较. 因为第 m 次迭代，或第 m 次重复这一过程需要 m 次比较，且 m 取值为 $p-1$，$p-2$，…，1. 因此，利用算术级数求和的标准公式，我们看到总共需要

$$(p-1)+(p-2)+\cdots+1=\frac{p(p-1)}{2}$$

步. 使用 2.18 节的术语，这一冒泡排序算法不如需要 $cp\log_2 p$ 步的算法有效. 因为冒泡排序是

209

$O(p^2)$算法，且 p^2 不是 $O(p\log_2 p)$.

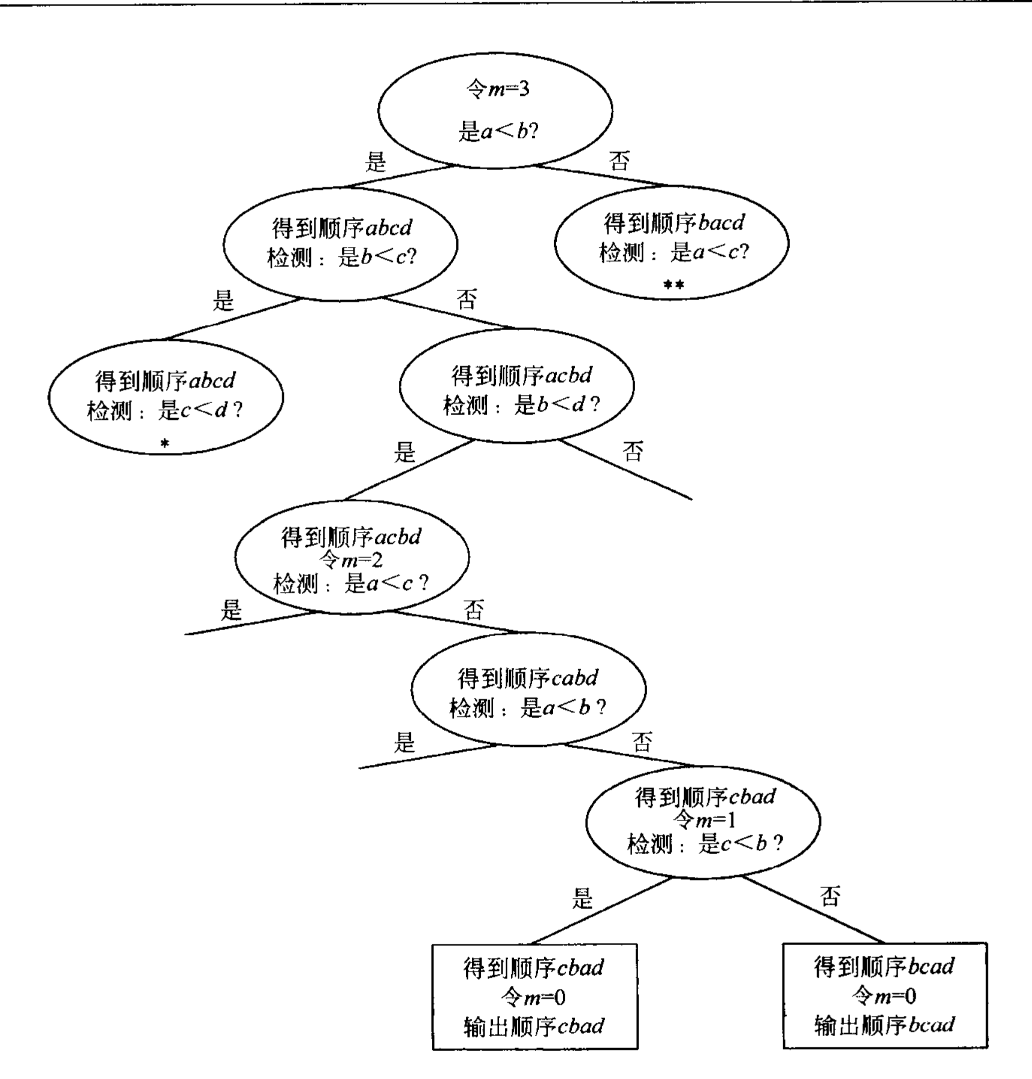


图 3.88 关于四个项 $abcd$ 的有序集合的冒泡排序的部分决策树
(标签 * 和 ** 用于练习 28)

210

3.6.5 完美种系发生问题

本节描述的模型不同于 3.5.7 节给出的对应于给定现存物种集合重构种系发生树的模型.

设 Σ 是一个字母表("字母"的集合). 一个 Σ 模式(pattern)是一个取自这一字母表的字母的序列. 例如, 如果 Σ 是由字母表的 26 个字母组成的, 那么 "tomato"是一个模式, 而如果 $\Sigma=\{0, 1\}$, 那么一个位串是一个模式. 设 P 是 Σ 的模式的一个有限集合. P 的一个关键字树(keyword tree)是满足下列条件的根树:

- 每一条边用一个来自于 Σ 的字母标签.
- 出自给定顶点的任意两条边有不同的标签.
- P 中的每一个模式正好对应于一个叶子, 使得从根到这一叶子的路径上的字母按其顺序给出这个模式.

作为例子, 图 3.89 给出模式集合{tomato, tomahawk, token, torrent, torpedo}的关键字树. 构建关键字树的算法在 Gusfield[1997]中有描述.

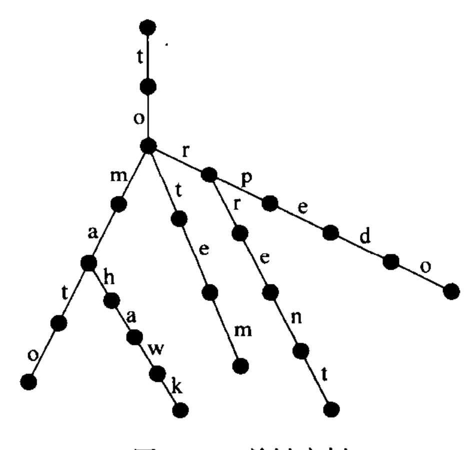


图 3.89 关键字树

重构进化历史的一个方法是：研究诸如物种这样的生物对象有或没有的特征．这些特征可能是诸如有脊椎、有羽毛或直立行走等等．它们还可能是更微妙的诸如“蛋白质 A 提高蛋白质 B 的表现”的事情，或者该类物种在 DNA 序列的第 p 个位置有给定的核苷（例如，腺嘌呤、A）等等．在所有这些情况中，我们说我们有**二分特征**(binary character)，其中，对象（特别是物种）或者有这一特征或者没有这一特征．假设我们感兴趣的是 n 个物种的

211

集合 S 和 m 个特征的集合 C．我们可以把进化考虑为从一位没有 C 中给定特征的祖先（树的根）开始发生的．在某一点，一个特征显现出来．于是，我们给一条从没有给定特征的给定顶点到它的孩子的边赋标签，其方法是：把这一特征的名字放置到这条边上．为简单起见，我们假设一旦一个特征显现出来，它就再不消失．于是，在我们得到这棵树的一个叶子时，从根到这一叶子的路径给出该物种所显现出来的所有特征的名字．图 3.90 给出一个例子．这里，物种 1 有特征 a 和 c，物种 3 有特征 a，e 和 d 等等．

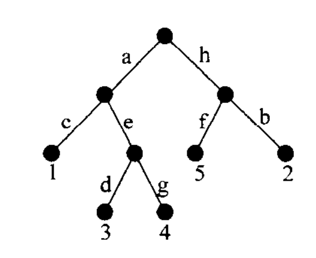


图 3.90 一棵种系发生树

已知每一物种拥有的特征的有关信息，实际上我们要重构类似于图 3.90 的树．为了精确地阐述这一问题，我们取 $\boldsymbol{M}$ 为 0 和 1 构成的 $n\times m$ 矩阵，它的 (i, j) 项等于 1 当且仅当物种 i 有特征 j．$\boldsymbol{M}$ 的**种系发生树**(phylogenetic tree)是一棵根树，它恰好有 n 个叶子和 m 条边，且满足下列性质：

- 集合 S 的 n 个物种中的每一个物种正好标记一个叶子．
- 集合 C 的 m 个特征的每一个特征正好标记一条边．
- 对于任意物种，沿着从根到以 i 为标签的叶子的路径上给边赋标签的特征指定了使得 $\boldsymbol{M}(i, j)=1$ 的所有特征 j．

我们希望知道，给定一个矩阵，是否能够为 $\boldsymbol{M}$ 构建一棵种系发生树．这就是**完美种系发生问题**(perfect phylogeny problem)．例如，考虑下面的矩阵 $\boldsymbol{M}$：

$$
\boldsymbol{M}=\begin{array}{c}\\ 1\\2\\3\\4\\5\end{array}\begin{array}{c}\begin{matrix} a & b & c & d & e & f & g & h\end{matrix}\\ \begin{bmatrix}1&0&1&0&0&0&0&0\\0&1&0&0&0&0&0&1\\1&0&0&1&1&0&0&0\\1&0&0&0&1&0&1&0\\0&0&0&0&0&1&0&1\end{bmatrix}\end{array}.
$$

对此，图 3.90 中的树给出 $\boldsymbol{M}$ 的一棵种系发生树．特别地，$\boldsymbol{M}$ 的种系发生树是一种由给定物种所具有的特征定义的模式集合的关键字树．并非每一个 $\boldsymbol{M}$ 都有种系发生树．例如，下面的矩阵就没有种系发生树：

212

$$
\begin{array}{c}\\ 1\\2\\3\\4\end{array}\begin{array}{c}\begin{matrix} a & b & c & d & e\end{matrix}\\ \begin{bmatrix}1&0&1&0&0\\1&1&0&1&0\\1&1&0&0&1\\0&1&0&0&0\end{bmatrix}\end{array}.
$$

我们从下面的定理将看到为什么这一矩阵没有种系发生树．然而，在看到下面的定理之前，读者也许希望使用这个小例子尝试着构建这样的一棵树．

为了解决完美种系发生问题，设 O_j 是在 $\boldsymbol{M}$ 的第 j 列上有 1 的物种的集合．

定理 3.28(Estabrook，Johnson 和 McMorris[1975，1976a，b])　一个矩阵有种系发生树当且仅当对于 j 和 k，或者 O_j 和 O_k 不相交，或者一个包含另一个.

使用 $O(mn)$ 步构建种系发生树的算法由 Gusfield[1991]给出. 更详细的内容可以参见 Gusfield[1997]. 如果每一个特征有两种以上的可能状态出现，这一问题就会变得更加复杂. 例如，我们可能说 DNA 链第 p 个位置能够有的状态是 A、T、C 或 G. 一些人研究了这种更一般情况下的完美种系发生问题，其中包括 Agarwala 和 Fernandez-Baca[1994]、Bodlaender，Fellows 和 Warnow[1992]、Kannan 和 Warnow[1994，1995]、Steel[1992].

本节练习

1. 对于图 3.91 中的根树，确定每一个顶点的层次.
2. 对于图 3.92 中的每棵根树，确定每一个顶点的层次.
3. 对于图 3.92 中的每棵根树，确定这棵树的高度.
4. 对于图 3.92 中的每棵根树，确定顶点 b 的所有后代.
5. 对于图 3.71 中的每棵树，标记每一个顶点，并为根选择一个顶点. 确定其余顶点的层次和这棵树的高度.
6. 图 3.91 中的树是二叉搜索树吗？如果是，描述如何搜索键 7.

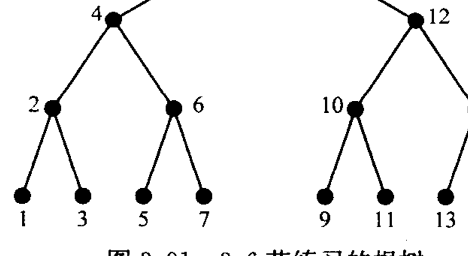
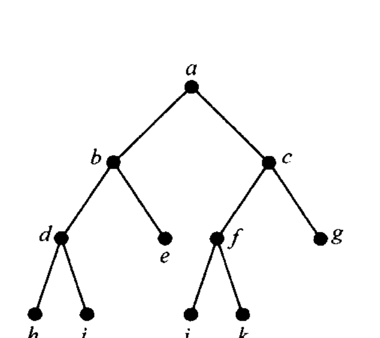
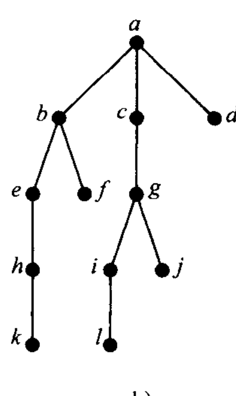


图 3.91　3.6 节练习的根树

7. 确定有 n 个顶点的平衡二叉树，其中

   (a)$n=5$　　(b)$n=8$　　(c)$n=12$　　(d)$n=15$





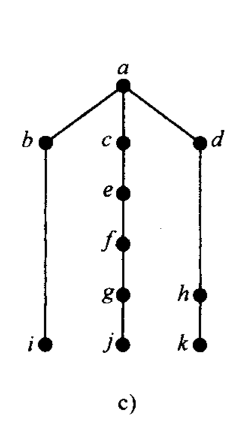


图 3.92　3.6 节练习的根树

8. 对于下列情况，确定一棵完全二叉树：

   (a)高度为 3 且 8 个叶子　　(b)高度为 3 和少于 8 个叶子

9. 确定有 11 个顶点且满足下列条件的完全二叉树：

   (a)高度尽可能大　　(b)高度尽可能小

10. 确定有 n 个顶点的二叉搜索树，其中 n 是：

    (a)10　　(b)14　　(c)18　　(d)20

11. 确定有 n 个顶点的二叉树的最小高度，其中 n 是：

    (a)74　　(b)512　　(c)4095

12. 完全二叉树可以用于编码 n 位的位串如下. 在任意顶点处，它的左孩子被标记为 0，它的右孩子被标记为 1. 于是一个位串对应于一个从根到一个没有孩子的顶点的简单链. 对于 $n=4$，画出一棵这样的树，并寻找对应于串 1011 的简单链.

13. 在有 n 个顶点的所有三叉(3 元)树中，最小的可能高度是多少？

213~214

14. 使用图 3.85 中的代码树，解码下列传输数据：

(a)0100100010110011　　(b)0001011100101000101110010l

15. 假设一个数据集合 D 是由四个花色的一组牌组成的.

(a)确定 D 的最长位串的长度为 2 的前辍码. 画出相关代码树.

(b)确定 D 的前辍码，它的一个位串长度是 3，且其余的更短. 画出相关的代码树.

16. 通过确定 g 和 h 的长度是 4 或小于 4 的位串来产生$\{a, b, c, d, e, f, g, h\}$的前辍码，其中下列的位串是已指定的：

a:00　b:011　c:10　d:1100　e:1101　f:111.

[提示：画出代码树.]

17. 使用算法 3.2，确定传输下列成绩的最优代码树：

(a)4 个 A，4 个 B，4 个 B−，4 个 C，4 个 C−，4 个 D，4 个 D−，4 个 F.

(b)8 个 A，6 个 A−，4 个 B+，1 个 B，5 个 B−，9 个 C+，11 个 C，5 个 C−，3 个 D+，3 个 D，5 个 D−，1 个 F.

18. 考虑下列数据：3 个 A，4 个 B，7 个 C，14 个 D，14 个 F

(a)证明算法 3.2 可以产生不同高度的最优代码树.

(b)你如何修改算法 3.2 使其产生最小高度的最优代码树.

19. 确定一个数据集合，对于这个数据集合，图 3.93 的树是最优代码树.

20. 这一练习证明算法 3.2 的输出的最优性.

215

(a)证明给定数据集合的最优代码树总是完全二叉树.

(b)证明算法 3.2 的输出对包含两个符号的数据集合是最优的.

(c)假设算法 3.2 的输出对于含有 $n-1$ 个符号，且 $n>2$ 的数据集合是最优的. 那么，给定一个包含 n 个符号的数据集合 D，除去带有最小频率(权)f_i 和 f_j 的两个符号 s_i 和 s_j，并用带有频率 $f_{ij}=f_i+f_j$ 的单一符号 s_{ij} 取代它们. 对于这样的新数据，你能说出算法 3.2 的输出是什么吗?

图 3.93　最优代码树

(d)给顶点 s_{ij} 附加分别有权 f_i 和 f_j 的两个孩子 s_i 和 s_j. 说明为什么这棵新树可以由算法 3.2 产生.

(e)解释为什么这棵新树对于 n 个符号的数据集合是最优代码树.

21. 假设我们有一棵*满平衡二叉搜索树*(fully balanced binary search tree)，即一棵平衡二叉搜索树(参见 137 页脚注)，其中每一个没有孩子的顶点都在同一层次上. 假设文件均等地处于 T 的 n 个顶点的任意一个顶点处. 如果我们使用寻找文件的平均步骤数量，而不是寻找文件的最大步骤数量来度量复杂度，那么使用 T 搜索文件的计算复杂度是什么?

22. 在一棵完全 m 元根树中，如果随机选取一个顶点，证明这一顶点有一个孩子的概率大约是 $1/m$.

23. (Tucker[1984])在编译器中，一个控制字被作为一个数存储. 假设可能的控制字是 GET，DO，ADD，FILL，STORE，REPLACE 和 WAIT，且这些控制字分别由数字 1、2、3、4、5、6、7 表示(用二进制符号). 给定一个未知的控制字 X，我们希望参照上述列表中的可能控制字检测它，直到我们找到它是哪个控制字. 一个方法是把 X 的数字依次与对应于控制字的数字 1、2、…、7 比较. 另一个方法是构建一棵二叉树. 描述后一种方法是如何工作的，并构建一棵这样的树.

24. *二分搜索算法*(binary search algorithm)搜索一个有序文件来查看一个键 x 是否在这一文件中. 这一文件的项是 n 个数，$x_1<x_2<\cdots<x_n$. 这一算法是把 x 与文件中间的第$\lceil n/2\rceil$个项 x_i 比较. 如果 $x=x_i$，这一搜索完毕. 如果 $x<x_i$，那么 $x_i, x_{i+1}, \cdots, x_n$ 被排除在考虑范围之外，这一算法搜索文件 $x_1, x_2, \cdots, x_{i-1}$，仍从中间项开始. 如果 $x>x_i$，那么 $x_1, x_2, \cdots, x_i$ 被排除在考虑范围之外，这一算法搜索文件 $x_{i+1}, x_{i+2}, \cdots, x_n$，仍从中间项开始. 反复迭代这一过程. 表 3.5 给出两个例子. (细节可以参见 Knuth[1973].)利用二分搜索算法在下列有序文件中搜索 150：

216

(a)22，36，83，148，150，190，201，225，216，221，301

(b)18，29，149，301，399，625，648，833，901，949

表 3.5 两个二分搜索

搜索 180												
71	97	164	180	285	436	513	519	522	622	663	687	用 180 与 436 比较
71	97	164	180	285								用 180 与 164 比较
			180	285								已经找到 180
搜索 515												
71	97	164	180	285	436	513	519	522	622	663	687	用 515 与 436 比较
						513	519	522	622	663	687	用 515 与 522 比较
						513	519					用 515 与 513 比较
							519					用 515 与 519 比较
												得到 515 不在文件中的结论

注：在每一种情况中，第一个列表包含整个文件，带下划线的数字是中间项，每一个子序列给出要搜索的剩余文件.

25. 计算练习 24 中定义的二分搜索算法的计算复杂度. [提示：证明该算法使用了一棵二叉搜索树.]
26. 对顺序 4312 使用冒泡排序，并说明使这一顺序变成自然顺序所需的步数.
27. 对下列顺序使用冒泡排序：
    (a)516324 (b)346152
28. 在图 3.88 中的冒泡排序的部分决策树中，填充开始于下列标签的顶点的部分：
    (a) * (b) **
29. 对应于下列各 p 的值，将最坏情况下冒泡排序算法的复杂度与定理 3.27 中的下界做比较：
    (a)7 (b)15
30. 在称为快速排序(quick sort)的排序算法中，我们开始于有 p 个项的一个有序列表，并尝试寻找其自然顺序. 我们从列表中选取第一个项，并把剩余项分成两组，即小于所选项的项为一组和大于这个项的项为一组. 然后，我们把所选项放在两组之间，且对每一组重复这一过程. 最终，我们得到只有一个元素组的各组，并且停止. 例如，给定顺序 31564827，我们选出项 3. 然后，3 之前的项的列表为 12，3 之后的项的列表为 56487. 现在，我们运用这一算法对这两组排序. 例如，从第二组中选出 5，得到两个子组 4 和 687. 现在，我们对它们排序. 以此类推. 运用快速排序算法对下列有序列表排序： 217
    (a)5176324 (b)941258376
31. 如果我们开始于列表 $123\cdots p$ 且 $p=5$，那么在最坏的情况下，快速排序算法(练习 30)需要多少步骤(多少次比较)?
32. 对于任意的 p 重复练习 31.
33. 对 h 作归纳证明定理 3.26.
34. 寻找对应于模式集合{sentry, seldom, spackle, spanking, spanning, seller}的关键字树.
35. 寻找对应于字母表 $\Sigma=\{A, T, G, C\}$ 上的模式集合{ATTCG, AATGC, ATGCC, AATTT, AATGTG}的关键字树.
36. 对于下列每一个由 0 和 1 构成的矩阵 $\boldsymbol{M}$，确定各完美种系发生问题是否有解，如果有解，则寻找 $\boldsymbol{M}$ 的种系发生树.

$$(a)\ \boldsymbol{M}=\begin{array}{c} \\ 1\\2\\3\\4 \end{array}\begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 0\\ 1 & 1 & 0 & 0\\ 1 & 0 & 1 & 0\\ 0 & 0 & 0 & 1 \end{bmatrix}\end{array}\qquad (b)\ \boldsymbol{M}=\begin{array}{c} \\ 1\\2\\3\\4\\5 \end{array}\begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 1\\ 1 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 1 & 0 & 1 & 0 & 0 \end{bmatrix}\end{array}\qquad (c)\ \boldsymbol{M}=\begin{array}{c} \\ 1\\2\\3\\4\\3 \end{array}\begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0\\ 1 & 1 & 1 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 1 & 0 & 1 & 0 \end{bmatrix}\end{array}$$

37. 证明：如果 0 和 1 构成的矩阵 M 有种系发生树，那么对于 j 和 k，或者 O_j 和 O_k 不相交或者一个包含另一个.

38. 在字典问题(dictionary problem)中，已知形成字典的模式集合. 当给出一个模式时，我们想知道它是否在字典中. 解释关键字树如何有助于解决这一问题?

39. 假设一个特征可以取状态 0，1，2，…，s 中的任意一个. 设 $\boldsymbol{M}$ 是一个项为$\{0, 1, 2, \cdots, s\}$的 $n\times m$ 矩阵，且如果物种 i 在状态 p 时有特征 j，则 $M(i, j)=p$. $\boldsymbol{M}$ 的一个*完美种系发生*(perfect phylogeny)是一棵根树，其中每一个物种正好标记一个叶子，且边用有序三元组(j, p, q)标记，意味着沿着这条边，特征 j 从状态 p 变到状态 q. 假设在根处，每一个特征的开始状态是给定的，而且对于$\{0, 1, 2, \cdots, s\}$中的任意特征 j 和 y，在从根到一个叶子的任意链上，至多有一条边对于某个 y 有形如(j, x, y)的标签(在这条链上，一个特征只可能变化到 y 一次).

(a)如果 $s=2$ 且 $\boldsymbol{M}=\begin{array}{c} \\ 1\\2\\3\\4\end{array}\begin{array}{c}\begin{array}{cc}a & b\end{array}\\ \begin{bmatrix}2&0\\1&2\\1&1\\2&2\end{bmatrix}\end{array}$，寻找 $\boldsymbol{M}$ 的一个完美种系发生.

218

(b)如果 $s=2$ 且 $\boldsymbol{M}=\begin{array}{c} \\ 1\\2\\3\\4\\5\end{array}\begin{array}{c}\begin{array}{ccc}a & b & c\end{array}\\ \begin{bmatrix}1&0&0\\2&2&2\\0&0&0\\0&1&2\\2&2&0\end{bmatrix}\end{array}$，寻找 $\boldsymbol{M}$ 的一个完美种系发生.

40. 假设 D 是一个有向图. 一个*层次分配*(level assignment)是一个层次 L_i 到每个顶点 i 的分配，使得如果(i, j)是一条弧，那么 $L_i<L_j$. 证明：有向图有层次分配当且仅当它没有循环.

41. 一个有向图 D 的*分级层次分配*(graded level assignment)是一个这样的层次分配(练习 40)，它使得如果 D 中存在一条弧(i, j)，那么有 $L_j=L_i+1$. 证明：如果 $D=(V, A)$有一个分级层次分配，那么它是*等路径的*(equipathic)；即对于 V 中所有的 u，v，所有从 u 到 v 的简单路径有相同的长度.

42. 假设 T 是一棵根树. 通过每一条边$\{u, v\}$从较低的层次指向较高的层次来给 T 赋方向. 其结果有向图称为*有向根树*(directed rooted tree).

(a)证明每棵有向根树恰好有一个顶点，从这一顶点开始其余每一个顶点都是通过一条路径可达的.

(b)有向根树是单连通的吗?

(c)证明每一棵有向有根树都有一个层次分配.

(d)假设 D 是一棵有向根树. 证明：D 有分级层次分配当且仅当 D 是等路径的.

43. 确定下列各情况的有根标签树的数量：

(a)五个顶点，有两个有度 2.

(b)五个顶点，其中根有度 2.

(c)四个顶点，其中根有度 1.

3.7　计算机中图的表示

图论中的很多问题只能在计算机上解决. 事实上，高速计算机器的发展已对图论问题的解决带来了重大的帮助，例如，四色问题，而且它也对图论在其他学科的应用带来重大的帮助. 与此同时，计算机科学的发展已给图论学家带来诸多有待解决的重要且富有挑战的问题. 本节讨论作为计算机程序的输入表示有向图或图的各种方法. 这也使得我们得以说明数学的另一个领域——矩阵理论对图论问题的帮助.

注意，我们使用过的有向图或图的图表对于较大的有向图或图不是非常实用，而且作为计算机的输入也不是很容易控制. 因此，需要其他输入有向图或图的方法. 输入有向图或图的最好

219

方法依赖于这一有向图或图的性质，以及有向图或图的用途．一个算法的有效性依赖于输入有向图或图的方法的选择，而且所需的内存存储也依赖于方法的选择．这里，我们给出几个输入有向图或图的不同方法．

我们集中讨论有向图的输入，因为图可以被认为是特殊的情况．把有向图 D 输入到计算机的最常用方法之一是给出它的**邻接矩阵**(adjacency matrix)$\mathbf{A}=\mathbf{A}(D)$．这一矩阵可以通过下面的方法得到：标记 D 的顶点为 $1, 2, \cdots, n$，并当在 D 中存在一条从 i 到 j 的弧时，令 $\mathbf{A}$ 的 i，j 项等于 1，当不存在这样的弧时，令 A 的 i，j 项等于 0．

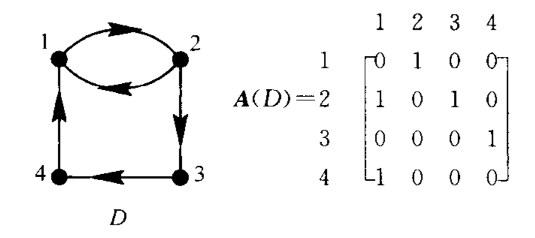


图 3.94 一个有向图和它的相关邻接矩阵 $\mathbf{A}(D)$

图 3.94给出一个有向图及与其相关的邻接矩阵[⊖]．注意，邻接矩阵可以作为一个二维数组或作为带有 n^2 个位的位串一行一行地输入．因此，需要 n^2 位的存储单元，如果对角线上的元素假设是 0，则需要$n(n-1)$位的存储单元，这是非循环有向图的情况．对于图，可以把它看成是对称有向图，存储需求更少，因为它的邻接矩阵是对称的．特别地，如果没有循环，那么我们需要

$$(n-1)+(n-2)+\cdots+1=\frac{n(n-1)}{2}$$

位的存储单元．因为我们只需要编码对角线上方的项．如果一个有向图是稀疏的，即它的弧很少，那么邻接矩阵不是一个非常有用的表示法．

另外一个对输入图(尽管不是有向图)有用的矩阵称为关联矩阵．一般地，假设 S 是一个集合，$\mathcal{F}$ 是 S 的子集的簇．如下定义**点集关联矩阵**(point-set incidence matrix)$\mathbf{M}$．标记 S 的元素为 $1, 2, \cdots, n$，设$\mathcal{F}$中的集合为 $S_1, S_2, \cdots, S_m$．那么 $\mathbf{M}$ 是一个 $n\times m$ 矩阵，如果元素 i 在集合 S_j 内，则它的 i，j 项是 1，如果元素 i 不在集合 S_j 内，则它的 i，j 项是 0．例如，如果$S=\{1, 2, 3, 4\}$且$\mathcal{F}=\{\{1, 2\}, \{2, 3\}, \{1, 4\}\}$，那么，$\mathbf{M}$ 由下面的矩阵给出．

220

$$\mathbf{M}=\begin{array}{c} \\ 1\\2\\3\\4\end{array}\begin{array}{c}\begin{array}{ccc}\{1,2\} & \{2,3\} & \{1,4\}\end{array}\\ \begin{bmatrix}1 & 0 & 1\\ 1 & 1 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\end{array}. \tag{3.26}$$

如果 G 是一个图，它的**关联矩阵**(incidence matrix)是集合 $S=V(G)$和集合$\mathcal{F}=E(G)$的点集关联矩阵．(3.26)式的矩阵 $\mathbf{M}$ 是图 3.95 中的图的关联矩阵．一个关联矩阵需要 $n\times e$ 位的存储单元，其中 e 是边的数目．这通常大于邻接矩阵所需的$\frac{1}{2}n(n-1)$位存储单元，而且对于连通图它总是至少需要 $n(n-1)=n^2-n$ 位的存储单元(为什么)．关联矩阵广泛用于电网和开关网络的输入．对有向图也可以定义关联矩阵(参见练习 20)．

3 2 1 4

图 3.95 由(3.26)式给出的关联矩阵的图

输入有向图 D 的另一个方法是简单地给出一个**弧列表**，即 D 的弧的列表．此时需要多少存储单元呢？假设每一个顶点标签通过长度为 t 的位串编码．那么如果 D 有 a 条弧，因为每一个弧被编码为一对码字，所以需要 $2at$ 位的存储单元．注意，t 必须足够大使得长度为 t 的位串的数量 2^t 至少是 n，而 n 是顶点个数．所以 $t\geqslant\log_2 n$．事实上，

$$t\geqslant\lceil\log_2 n\rceil,$$

⊖ 当然，这一特殊的矩阵依赖于我们列出项点的方法．但是，我们简单地称任何一个这样的矩阵为邻接矩阵．

其中$\lceil x\rceil$是大于或等于x的最小整数. 因此，我们至少需要$2a\lceil\log_2 n\rceil$位的存储单元. 如果弧不是很多，也就是说，如果邻接矩阵是稀疏的，那么这个数可能比n^2小，所以需要的存储少于邻接矩阵的存储.

有两个弧列表的变形. 第一个变形是输入两个线性序或数组，每一个都有a个码字，其中a是弧集合的大小. 如果这些数组是$(h_1, h_2, \cdots, h_a)$和$(k_1, k_2, \cdots, k_a)$，且h_i编码顶点x，k_i编码顶点y，那么(x, y)是弧列表中第i个弧. 所需的存储单元与弧列表所需的存储单元一样多.

弧列表的另一个变形是给出n个邻接列表(adjacency list)，D的每一个顶点x有一个邻接列表，该列表列出使(x, y)为弧的顶点y. 这n个列表称为一个邻接结构(adjacency structure). 一个邻接结构需要$(n+a)t$位的存储单元，其中t与弧列表中t相同. 这是因为顶点x的邻接列表必须通过编码x然后编码所有与x邻接的y而进行编码.

正如我们已指出的那样，向计算机输入有向图或图的最好方法取决于我们使用这一输入的目的. 作为一个简单例子，假设我们问：(u_i, u_j)是弧吗？这一问题在邻接矩阵存储可以一步就给出答案(检查i，j项). 在邻接结构存储回答这一问题在最坏情况下需要$\deg(u_i)$步，如果u_j是从u_i出发的一条弧达到的最后一个顶点. 另一方面，让我们考虑需要标记从u_i出发的一条弧能达到的所有顶点的算法. 在邻接矩阵存储这需要n步(检查第i行的所有项)，但在邻接结构存储只需要$\deg(u_i)$步.

221

本节练习

1. 对于图 3.7 中的每一个有向图，确定：
   (a)它的邻接矩阵.
   (b)一个弧列表.
   (c)可以用于输入弧的两个线性数组.
   (d)一个邻接结构.
2. 对于图 3.23 中的每一个图，确定一个关联矩阵.
3. 画出邻接矩阵由下列矩阵所给出的有向图

$$(a)\mathbf{A}=\begin{bmatrix}0&0&0&1&1\\1&0&0&0&0\\1&0&0&1&0\\0&1&0&0&1\\1&1&1&0&0\end{bmatrix}\qquad (b)\mathbf{A}=\begin{bmatrix}0&1&0&0&1&0\\0&0&1&0&0&1\\0&0&0&1&0&0\\1&0&0&0&1&0\\0&1&0&0&0&1\\1&0&0&0&0&0\end{bmatrix}$$

4. 确定有向图，其顶点为a，b，c，d，e，f且弧列表是
$$\{(b,a),(f,a),(b,d),(e,c),(e,b),(c,a),(c,e),(a,d)\}.$$
5. 确定有向图，其顶点为a，b，c，d，e，且其弧由下面两个线性数组(a, b, c, c, d, d, e)，(b, d, a, d, c, e, a)编码.
6. 确定有向图，其邻接结构由表 3.6 给出.

表 3.6　一个有向图的邻接结构

顶点x	1	2	3	4	5	6
邻接于x的顶点	2，4	1，3，4	2	1，5，6	3	

7. 计算图 3.9 中的每一个图的邻接矩阵(这一邻接矩阵是所对应的有向图的邻接矩阵).
8. 确定图 3.76 中的图G的点集关联矩阵，如果：

(a)$S=V(G)$，且$\mathcal{F}=G$的所有支撑树.

(b)$S=E(G)$，且$\mathcal{F}=G$的所有支撑树.

222

9. 如果D^c是D的补有向图(3.2 节的练习 34)，那么$\boldsymbol{A}(D)+\boldsymbol{A}(D^c)$是什么？

10. 假设扫描一个列表或数组中的一个项需要一步. 假设一个有向图存储为一个邻接矩阵.

(a)标记所有与特殊顶点y邻接的顶点x，即使得(x, y)是弧，需要多少步？

(b)标记或计数所有弧需要多少步？

11. 如果有向图存储为弧列表，重复练习 10.

12. 如果有向图存储为邻接结构，重复练习 10.

13. 如果D是一个有n个顶点的有向图，它的可达矩阵(reachability matrix)是一个$n\times n$矩阵$\boldsymbol{R}$，如果顶点j是从顶点i通过一条路径可达的，则它的i，j项r_{ij}是 1，否则是 0. 对于图 3.7 中的每一个有向图，确定它的可达矩阵(注意i总是从i可达的).

14. D的可达矩阵$\boldsymbol{R}(D)$和$\boldsymbol{R}(D^c)$之间是什么关系？

15. 证明：D是强连通的当且仅当它的可达矩阵(练习 13)是所有项都是 1 的矩阵$\boldsymbol{J}$.

16. 如果D是邻接矩阵为$\boldsymbol{A}$的有向图，对k作归纳证明：$\boldsymbol{A}^k$的i，j项给出D中从i到j的长度为k的路径的数量.

17. 如果G是邻接矩阵为$\boldsymbol{A}$(练习 7)的图，用图(与有向图相对)语言，解释$\boldsymbol{A}^k$(练习 16)的i，j项是什么？

18. 假设 0-1 对称方阵在其对角线上有 0. 它一定是某个图的邻接矩阵吗？

19. 对于图 3.7 中的有向图D_1，D_2和D_7，使用练习 16 的结果确定从u到v的长度为 3 的路径的数量. 确认这些路径.

20. 如果D是一个有向图，它的关联矩阵其行对应于顶点，列对应于弧，且如果对某个k，j是弧(i, k)，则i，j项等于 1，如果对于某个k，j是弧(k, i)，则是-1，否则是 0.

(a)确定图 3.7 中的每一个有向图的关联矩阵.

(b)对于这一关联矩阵需要多少位存储单元？

(c)如果$\boldsymbol{M}=(m_{ij})$是有向图D的关联矩阵，矩阵$\boldsymbol{N}=(n_{ij})$，其中$n_{ij}=\sum_k m_{ik}m_{jk}$的意义是什么？

(d)如果D是强连通的，我们能否成功地比这一邻接矩阵使用更少的存储位数？

21. 如果$\boldsymbol{M}=(m_{ij})$是非负项的任意矩阵，设$B(\boldsymbol{M})$是这样的一个矩阵：如果$m_{ij}>0$，则它的i，j项是 1，如果$m_{ij}=0$，则是 0. 证明：如果D是有n个顶点的有向图，且有可达矩阵$\boldsymbol{R}$和邻接矩阵$\boldsymbol{A}$，$\boldsymbol{I}$是单位矩阵，那么

(a)$\boldsymbol{R}=B[\boldsymbol{I}+\boldsymbol{A}+\boldsymbol{A}^2+\cdots+\boldsymbol{A}^{n-1}]$　　(b)$\boldsymbol{R}=B[(\boldsymbol{I}+\boldsymbol{A})^{n-1}]$

22. 对于图 3.7 中的图D_1和D_7核实练习 21 的结果.

223

23. (a)证明：D是单连通的(3.2 节练习 9)当且仅当$B(\boldsymbol{R}+\boldsymbol{R}^{\mathrm{T}})=\boldsymbol{J}$，其中$B$是在练习 21 中所定义的，$\boldsymbol{R}$是在练习 13 中所定义的，$\boldsymbol{J}$是在练习 15 中所定义的，而$\boldsymbol{R}^{\mathrm{T}}$是$\boldsymbol{R}$的转置.

(b)证明：D是弱连通的(3.2 节练习 10)当且仅当$B[(\boldsymbol{I}+\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}})^{n-1}]=\boldsymbol{J}$.

24. 假设D是一个有向图，有可达矩阵(练习 13)$\boldsymbol{R}=(r_{ij})$且$\boldsymbol{R}^2$是矩阵(s_{ij}). 证明：

(a)包含顶点i的强分支(3.2 节练习 13)由在$\boldsymbol{T}=(t_{ij})$的第i行上为 1 的项给出，其中$t_{ij}=r_{ij}\times r_{ij}^{(\mathrm{T})}$，而$r_{ij}^{(\mathrm{T})}$是$\boldsymbol{R}$的转置的$i$，$j$项.

(b)包含i的强分支中的顶点数量是s_{ii}.

25. 对于图 3.7 中的每一个有向图，使用练习 24 的结果确定强分支.

26. 如果$\boldsymbol{R}$是一个有向图的可达矩阵且$c(i)$是$\boldsymbol{R}$的第i列的和，解释$c(i)$是什么.

27. 如果G是一个图，你将如何直接定义它的可达矩阵$\boldsymbol{R}(G)$？

28. 如果G是一个图，$\boldsymbol{R}=\boldsymbol{R}(G)$是它的可达矩阵(练习 27)，且$\boldsymbol{T}$的定义如练习 24：

(a)证明$\boldsymbol{T}=\boldsymbol{R}$.

(b)解释$\boldsymbol{R}^2$的 1，1 项是什么.

29. 假设 $\boldsymbol{R}$ 是 0 和 1 的矩阵，且对角线的下方都是 1(其他地方也可能是 1). $\boldsymbol{R}$ 一定是某个有向图的可达矩阵吗？(给出证明或举出反例.)
30. (Harary[1969])假设 $\boldsymbol{B}$ 是图 G 的关联矩阵，且 $\boldsymbol{B}^{\mathrm{T}}$ 是 $\boldsymbol{B}$ 的转置. 矩阵 $\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}$ 的 i，j 项的意义是什么？
31. (Harary[1969])设 G 是一个图. G 的**回路矩阵**(circuit matrix)$\boldsymbol{C}$ 是 G 的边集合 S 和 G 的回路簇 $\mathcal{F}$ 的点集关联矩阵. 设 $\boldsymbol{B}$ 是 G 的关联矩阵. 证明 $\boldsymbol{BC}$ 的每一个项模 2 等于 0.
32. 两个图能够有相同的关联矩阵且是非同构的吗？为什么？
33. 两个图能够有相同的回路矩阵且是非同构的吗？为什么？如果每一条边都在一个回路上时，情况又如何？

3.8　再论拉姆齐数

我们在 2.19.3 节介绍了拉姆齐理论，特别是拉姆齐数 $R(p, q)$. 为了研究拉姆齐数，使用图论是很便利的. 如果 G 是一个图，它的补(complement)G^c 是一个与 G 有相同顶点集合，且使得对于 $V(G)$ 中所有 $a\neq b$，有 $\{a, b\}\in E(G^c)$ 当且仅当 $\{a, b\}$ 不属于 $E(G)$ 的图. 在研究拉姆齐数 $R(p, q)$ 的过程中，我们把一个集合 S 作为一个图的顶点集合，S 的 2 元素子集为这个图的边集合. 这时，说一个数 N 有 (p, q) 拉姆齐性质的意思是：对于任意的 N 元素集合 S 和以 S 为顶点集合的图 G，如果我们把 S 的 2 元素集分成 G 的边和 G^c 的边(不在 G 中的边)，那么或者存在 p 个顶点，它们都由 G 中的边相互连接，或者存在 q 个顶点，它们都由 G^c 中的边相互连接. 换一种方式，我们有下面的定理.

224

定理 3.29　*一个数 N 有 (p, q) 拉姆齐性质当且仅当对于 N 个顶点的每一个图 G，或者 G 有一个完全 p 边形(p 个顶点的完全子图，K_p)或者 G^c 有一个完全 q 边形.*

推论 3.29.1　$R(p, 2)=p$ 且 $R(2, q)=q$.

证明　对于有 p 个顶点的每一个图，或者它是完全的，或者它的补有一条边. 这就证明 $R(p, 2)\leqslant p$. 显然 $R(p, 2)\geqslant p$(为什么). ■

使用定理 3.29 的术语，定理 2.18 说的是如果 G 是 6(或更多)个顶点的图，那么或者 G 有一个三角形或者 G^c 有一个三角形(读者应该选择一些图来进行验证). 更一般地，(至少有)$R(p, q)$ 个顶点的每一个图或者有一个完全 p 边形(K_p)或者它的补有一个完全 q 边形(K_q).

现在，考虑图 $G=Z_5$，长度为 5 的回路. 现在，G^c 也是(同构于)Z_5. 因此，G 和 G^c 都没有三角形. 我们得出结论：5 没有 (3, 3) 拉姆齐性质，所以 $R(3, 3)>5$. 这完成了下面定理的证明.

定理 3.30　$R(3, 3)=6$.

使用 3.3.1 节的术语，G 中的一个完全 p 边形是有 p 个顶点的一个团. G^c 中的一个完全 q 边形对应于 G 中有 q 个顶点的独立集合. 因此，我们可以重述定理 3.29 如下.

定理 3.31　*一个数 N 有 (p, q) 拉姆齐性质，当且仅当对于有 N 个顶点的每一个图 G，或者 G 有 p 个顶点的团或者 G 有 q 个顶点的独立集合.*

还有另一个重述定理 3.29 的方式，如下所示.

定理 3.32　*一个数 N 有 (p, q) 拉姆齐性质，当且仅当用红色和蓝色着色有 N 个顶点的完全图 K_N 的边，K_N 有一个完全红色的 p 边形或 K_N 有一个完全蓝色的 q 边形.*

证明　给定一个边着色，令 G 是这样的图，它的顶点与 K_N 的顶点相同，其边是红色边. ■

我们已观察到拉姆齐数是很难计算的. 已确定的少数几个拉姆齐数如表 2.12 所示. 为了证实(至少部分证实)表 2.12 中的一些数，你能够得到拉姆齐数的某些边界. 考虑图 3.96 中的图 a.

225

这个图有 8 个顶点. 它即没有三角形(3 边形)，而且也没有有 4 个顶点的独立集合. 因此，根据定理 3.31，数 8 没有(3，4)拉姆齐性质. 因此有 $R(3，4)\geqslant 9$. 类似地，图 3.96中的图 b 和 c 证明 $R(3，5)\geqslant 14$，且 $R(4，4)\geqslant 18$(练习 2 和 3).

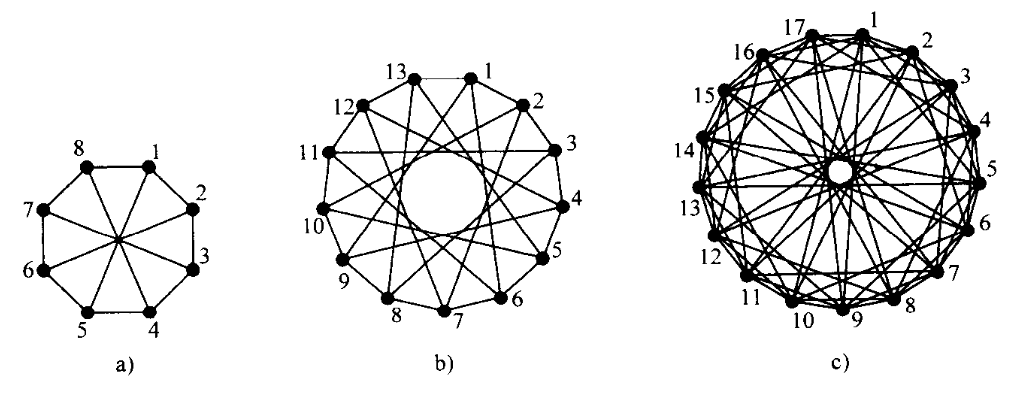


图 3.96 说明 a)$R(3，4)\geqslant 9$，b)$R(3，5)\geqslant 14$，c)$R(4，4)\geqslant 18$ 的图

最后，拉姆齐数的变形给图论带来相当困难的挑战. 对于其中关于图拉姆齐数的课题的讨论，可以参见练习 11 到练习 13.

本节练习

1. 证明 $R(p，2)\geqslant p$.
2. 使用图 3.96b 证明 $R(3，5)\geqslant 14$.
3. 使用图 3.96c 证明 $R(4，4)\geqslant 18$.
4. 设 G 是一个有 25 个顶点的完全图，且令 G 的边或者被着色成棕色或被着色成绿色. 如果不存在绿色三角形，那么你能确定 G 的最大的完全棕色 m 边形是什么吗?
5. 对于图 3.97 中的每一个图，或者确定一个有 3 个顶点的团，或者确定有 3 个顶点的独立集合，或者得出它们均不存在的结论.

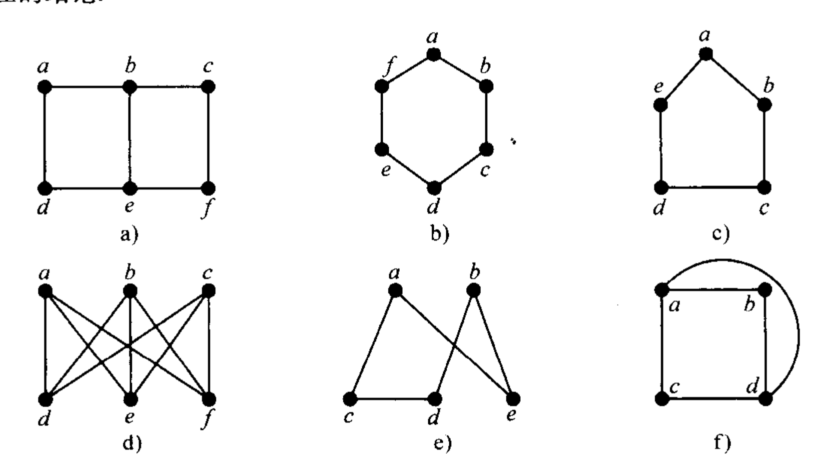


图 3.97 3.8 节练习 5 的图

6. 令 G 是有 11 个顶点且色数是 3 的任意图.
   (a)G 一定或者有 3 个顶点的团或者有 3 个顶点的独立集合吗?
   (b)G 一定或者有 4 个顶点的团或者有 3 个顶点的独立集合吗?
7. 设 G 是有 16 个顶点且最大团的大小是 3 的图.
   (a)G 一定有 4 个顶点的独立集合吗?

(b)G一定有5个顶点的独立集合吗?

8. 用红色和蓝色着色K_{10}的边.
   (a)证明：如果至少从一个顶点出发存在4条红色边，那么或者存在全部由红色边连接的3个顶点，或者存在全部由蓝色边连接的4个顶点.
   (b)类似地，如果至少从一个顶点出发存在6条蓝色边，证明或者存在全部由红色边连接的3个顶点，或者存在全部由蓝色边连接的4个顶点.
   (c)证明：根据(a)和(b)，使用红色和蓝色着色K_{10}的边，或者存在全部由红色边连接的3个顶点，或者存在全部由蓝色边连接的4个顶点.
   (d)关于拉姆齐数部分(c)告知你什么?
9. 设G是有20个顶点的树.
   (a)G一定有5个顶点的独立集合吗?
   (b)G一定有6个顶点的独立集合吗?
10. 使用红色、白色和蓝色着色图K_{17}的边. 这一练习将证明存在全部由相同颜色的边连接的3个顶点.
   (a)固定一个顶点a. 证明：连接这一顶点的所有边中，至少有6条边有相同的颜色.
   (b)假设(a)中的6条边都是红色. 这些边是从a到6个顶点b，c，d，e，f，g的边. 由此证明K_{17}有一个红色三角形、一个蓝色三角形或一个白色三角形.
   (c)关于2.19节的练习33定义的拉姆齐数$R(p, q; r)$，这一结果说明了什么?

227

11. 设G_1和G_2是图. 一个整数N有图拉姆齐性质(graph Ramsey property)(G_1，G_2)，如果使用颜色1和2的完全图K_N的每一个边着色，对于某个i，生成一个(同构于)G_i的子图且全部用i着色，即对于$i=1$或2，生成一个单色调G_i. 图拉姆齐数(graph Ramsey number)$R(G_1, G_2)$是有图拉姆齐性质(G_1，G_2)的最小的N(不难证明这是良好定义的. 参见Chartrand和Lesniak[1996]、Graham，Rothschild和Spencer[1990]). 如果L_p是p个顶点的链，且Z_q是q个顶点的回路：
   (a)证明$R(L_3, L_3)=3$.
   (b)证明$R(L_4, L_4)=5$.
   (c)确定$R(L_3, L_4)$.
   (d)确定$R(L_3, Z_4)$.
   (e)确定$R(L_4, Z_4)$.
   (f)确定$R(Z_4, Z_4)$.
12. (Chvátal和Harary[1972])令$c(G)$是G的最大连通分支的大小. 证明：
$$R(G,H) \geqslant (\chi(G)-1)(c(H)-1)+1.$$
13. (Chvátal[1977])如果T_m是有m个顶点的树，证明：
$$R(T_m,K_n) = 1+(m-1)(n-1).$$

参考文献

Agarwala, R., and Fernandez-Baca, D., "A Polynomial-time Algorithm for the Perfect Phylogeny Problem When the Number of Character States Is Fixed," *SIAM J. Comput.*, *23* (1994), 1216–1224.

Agarwala, R., Bafna, V., Farach, M., Narayanan, B., Paterson, M., and Thorup, M., "On the Approximability of Numerical Taxonomy: Fitting Distances by Tree Metrics," *Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 1996.

Aho, A. V., Hopcroft, J. E., and Ullman, J. D., *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B., *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, NJ, 1993.

ALON, N., "Restricted Colorings of Graphs," in K. Walker (ed.), *Surveys in Combinatorics*, London Mathematical Society Lecture Note Series, 187, Cambridge University Press, Cambridge, 1993, 1–33.

APPEL, K., and HAKEN, W., "Every Planar Map Is Four Colorable. Part I: Discharging," *Ill. J. Math., 21* (1977), 429–490.

APPEL, K., HAKEN, W., and KOCH, J., "Every Planar Map Is Four Colorable. Part II: Reducibility," *Ill. J. Math., 21* (1977), 491–567.

BAASE, S., *Computer Algorithms*, 2nd ed., Addison-Wesley Longman, Reading, MA, 1992.

BALABAN, A. T. (ed.), *Chemical Applications of Graph Theory*, Academic Press, New York, 1976.

BELTRAMI, E. J., and BODIN, L. D., "Networks and Vehicle Routing for Municipal Waste Collection," *Networks, 4* (1973), 65–94.

228

BERGE, C., "Färbung von Graphen, deren sämtliche bzw. deren ungerade Kreise starr sind," *Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg, Math.-Naturwiss. Reihe, 10* (1961), 114.

BERGE, C., "Sur une conjecture relative au problème des codes optimaux," *Commun. 13ème Assemblèe Gènèrale de l'URSI* (International Scientific Radio Union), Tokyo, 1962.

BERN, M. W., and GRAHAM, R. L., "The Shortest-Network Problem," *Scientific American, 260* (1989), 84–89.

BIRKHOFF, G. D., "A Determinant Formula for the Number of Ways of Coloring a Map," *Ann. Math., 14* (1912), 42–46.

BIRKHOFF, G. D., and LEWIS, D. C., "Chromatic Polynomials," *Trans. Amer. Math. Soc., 60* (1946), 355–451.

BODLAENDER, H. L., "Polynomial Algorithms for Graph Isomorphism and Chromatic Index on Partial k-Trees," *J. Algorithms, 11* (1990), 631–643.

BODLAENDER, H. L., FELLOWS, M., and WARNOW, T., "Two Strikes Against Perfect Phylogeny," *Proc. of the 19th Int. Colloq. on Automata, Languages, and Programming*, (1992), 273–283.

BONDY, J. A., and MURTY, U. S. R., *Graph Theory with Applications*, Elsevier, New York, 1976.

BONIAS, I., "T-colorings of Complete Graphs," Ph.D. thesis, Northeastern University, 1991.

BOOTH, K. S., and LUEKER, G. S., "Testing for the Consecutive Ones Property, Interval Graphs, and Graph Planarity Using P Q-tree Algorithms," *J. Comp. Syst. Sci., 13* (1976), 335–379.

BRASSARD, G., and BRATLEY, P., *Algorithmics: Theory and Practice*, Prentice Hall, Upper Saddle River, NJ, 1995.

BRIGHAM, R. C., and DUTTON, R. D., "Generalized k-Tuple Colorings of Cycles and Other Graphs," *J. Comb. Theory, Series B, 32* (1982), 90–94.

BUNEMAN, P., "The Recovery of Trees from Measures of Dissimilarity," in F. R. Hodson, D. G. Kendall, and P. Tautu (eds.), *Mathematics in the Archaeological and Historical Sciences*, Edinburgh University Press, Edinburgh, 1971, 387–395.

CAYLEY, A., "On the Theory of the Analytical Forms Called Trees," *Philos. Mag., 13* (1857), 172–176. [Also *Math. Papers*, Cambridge, *3* (1891), 242–246.]

CAYLEY, A., "On the Mathematical Theory of Isomers," *Philos. Mag., 67* (1874), 444–446. [Also *Math. Papers*, Cambridge, *9* (1895), 202–204.]

CAYLEY, A., "A Theorem on Trees," *Quart. J. Math., 23* (1889), 376–378. [Also *Math. Papers*, Cambridge, *13* (1897), 26–28.]

CHARTRAND, G., and LESNIAK, L., *Graphs and Digraphs*, 3rd ed., CRC Press, Boca Raton, 1996.

CHUDNOVSKY, M., ROBERTSON, N., SEYMOUR, P. D., and THOMAS, R., "The Strong Perfect Graph Theorem," manuscript, (2002).

CHVÁTAL, V., "Tree-Complete Graph Ramsey Numbers," *J. Graph Theory, 1* (1977), 93.

CHVÁTAL, V., and HARARY, F., "Generalized Ramsey Theory for Graphs," *Bull. Amer. Math. Soc., 78* (1972), 423–426.

CIESLIK, D., *Steiner Minimal Trees*, Kluwer Academic Publishers, Norwell, MA, 1998.

COHEN, J. E., *Food Webs and Niche Space*, Princeton University Press, Princeton, NJ, 1978.

COHEN, J., and FARACH, M., "Numerical Taxonomy on Data: Experimental Results," *J. Comput. Biol., 4* (1997), 547–558.

229

COZZENS, M. B., and ROBERTS, F. S., "T-Colorings of Graphs and the Channel Assignment Problem," *Congressus Numerantium, 35* (1982), 191–208.

DEMOURCRON, G., MALGRANCE, V., and PERTUISET, R., "Graphes planaires: reconnaissance et construction des représentations planaires topologiques," *Rev. Française Recherche Opérationnelle, 8* (1964), 33–47.

DEO, N., *Graph Theory with Applications to Engineering and Computer Science*, Prentice Hall, Englewood Cliffs, NJ, 1974.

ERDŐS, P., RUBIN, A. L., and TAYLOR, H., "Choosability in Graphs," *Congressus Numerantium, 26* (1979), 125–157.

ERDŐS, P. L., STEEL, M. A., SZÉKELY, L. A., and WARNOW, T. J., "Constructing Big Trees from Short Sequences," in P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela (eds.), *ICALP'97, 24th International Colloquium on Automata, Languages, and Programming (Silver Jubilee of EATCS)*, Bologna, Italy, July 7-11, 1997, *Lecture Notes in Computer Science*, Vol. 1256, Springer-Verlag, Berlin, 1997, 827–837.

ERDŐS, P. L., STEEL, M. A., SZÉKELY, L. A., and WARNOW, T. J., "A few logs suffice to build (almost) all trees. I," *Random Structures Algorithms, 14* (1999), 153–184.

ESTABROOK, G., JOHNSON, C., and MCMORRIS, F. R., "An Idealized Concept of the True Cladistic Character," *Math. Bioscience, 23* (1975), 263–272.

ESTABROOK, G., JOHNSON, C. and MCMORRIS, F. R., "A Mathematical Foundation for the Analysis of Cladistic Character Compatibility," *Math. Bioscience, 29* (1976), 181–187. (a)

ESTABROOK, G., JOHNSON, C., and MCMORRIS, F. R., "An Algebraic Analysis of Cladistic Characters," *Discrete Math., 16* (1976), 141–147. (b)

EULER, L., "Solutio Problematis ad Geometriam Situs Pertinentis," *Comment. Acad. Sci. 1 Petropolitanae, 8* (1736), 128–140. [Reprinted in *Opera Omnia*, Series 1–7 (1766), 1–10.]

EVEN, S., *Graph Algorithms*, Computer Science Press, Potomac, MD, 1979.

FILLIBEN, J. J., KAFADAR, K., and SHIER, D. R., "Testing for Homogeneity of Two-Dimensional Surfaces," *Math Modeling, 4* (1983), 167–189.

FIORINI, S., and WILSON, R. J., *Edge Colorings of Graphs*, Pitman, London, 1977.

FITCH, W. M., "An Introduction to Molecular Biology for Mathematicians and Computer Programmers," in M. Farach-Colton, F. S. Roberts, M. Vingron, and M. S. Waterman (eds.), *Mathematical Support for Moleclar Biology*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 47, American Mathematical Society, Providence, RI, 1999, 1–31.

FRANK, A., "Connectivity and Network Flows," in R. L. Graham, M. Grötschel, and L. Lovász (eds.), *Handbook of Combinatorics*, Elsevier, Amsterdam, 1995, 111–177.

GAREY, M. R., and JOHNSON, D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, 1979.

GELLER, D. P., "r-Tuple Colorings of Uniquely Colorable Graphs," *Discrete Mathematics, 16* (1976), 9–12.

GIBBONS, A., *Algorithmic Graph Theory*, Cambridge University Press, Cambridge, 1985.

GILBERT, E. N., Unpublished Technical Memorandum, Bell Telephone Labs, Murray Hill, NJ, 1972.

GOLDMAN, A. J., "Discrete Mathematics in Government," lecture presented at SIAM

Symposium on Applications of Discrete Mathematics, Troy, NY, June 1981.

GOLUMBIC, M. C., *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980. [230]

GRAHAM, R. L., and HELL, P., "On the History of the Minimum Spanning Tree Problem," *Annals of the History of Computing, 7* (1985), 43–57.

GRAHAM, R. L., ROTHSCHILD, B. L., and SPENCER, J. H., *Ramsey Theory*, 2nd ed., Wiley, New York, 1990.

GRAVIER, S., "Coloration et Produits de Graphes," Ph.D. thesis, Université Joseph Fourier, Grenoble, France, 1996.

GRIMALDI, D. A., *A Phylogenetic, Revised Classification of Genera in the Drosophilidae (Diptera)*, Bulletin of the American Museum of Natural History, American Museum of Natural History, New York, 1990.

GUSFIELD, D., "Efficient Algorithms for Inferring Evolutionary History," *Networks, 21* (1991), 19–28.

GUSFIELD, D., *Algorithms on Strings, Trees, and Sequences*, Cambridge University Press, New York, 1997.

HALE, W. K., "Frequency Assignment: Theory and Applications," *Proc. IEEE, 68* (1980), 1497–1514.

HALIN, R., "Bemerkungen Über Ebene Graphen," *Math. Ann., 53* (1964), 38–46.

HANSEN, P., FOWLER, P., and ZHENG, M. (eds.), *Discrete Mathematical Chemistry*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 51, American Mathematical Society, Providence, RI, 2000.

HARARY, F., *Graph Theory*, Addison-Wesley, Reading, MA, 1969.

HARARY, F., NORMAN, R. Z., and CARTWRIGHT, D., *Structural Models: An Introduction to the Theory of Directed Graphs*, Wiley, New York, 1965.

HARARY, F., and PALMER, E. M., *Graphical Enumeration*, Academic Press, New York, 1973.

HARARY, F., and TUTTE, W. T., "A Dual Form of Kuratowski's Theorem," *Canad. Math. Bull., 8* (1965), 17–20.

HARRISON, J. L., "The Distribution of Feeding Habits among Animals in a Tropical Rain Forest," *J. Anim. Ecol., 31* (1962), 53–63.

HOPCROFT, J. E., and TARJAN, R. E., "Efficient Planarity Testing," *J. ACM, 21* (1974), 549–568.

HUFFMAN, D. A., "A Method for the Construction of Minimum Redundancy Codes," *Proc. Inst. Rail. Eng., 40* (1952), 1098–1101.

IRVING, R. W., "NP-completeness of a Family of Graph-Colouring Problems," *Discrete Appl. Math., 5* (1983), 111–117.

JENSEN, T. R., and TOFT, B., *Graph Coloring Problems*, Wiley, New York, 1995.

KANG, A. N. C., LEE, R. C. T., CHANG, C. L., and CHANG, S. K., "Storage Reduction through Minimal Spanning Trees and Spanning Forests," *IEEE Trans. Comput., C-26* (1977), 425–434.

KANNAN, S., and WARNOW, T., "Inferring Evolutionary History from DNA Sequences," *SIAM J. Comput., 23* (1994), 713–737.

KANNAN, S., and WARNOW, T., "A Fast Algorithm for the Computation and Enumeration of Perfect Phylogenies When the Number of Character States Is Fixed," *6th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 1995, 595–603.

KELLY, J. B., and KELLY, L. M., "Paths and Circuits in Critical Graphs," *Amer. J. Math., 76* (1954), 786–792.

KEMENY, J. G., and SNELL, J. L., *Mathematical Models in the Social Sciences*, Blaisdell, New York, 1962. (Reprinted by MIT Press, Cambridge, MA, 1972.)

KIRCHHOFF, G., "Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Verteilung galvanischer Ströme geführt wird," *Ann. Phys. Chem., 72* (1847), 497–508. [231]

KLOTZ, W., "A Constructive Proof of Kuratowski's Theorem," *Ars Combinatoria, 28* (1989), 51–54.

KNUTH, D. E., *The Art of Computer Programming*, Vol. 3: *Sorting and Searching*, Addison-Wesley, Reading, MA, 1973.

KÖNIG, D., *Theorie des endlichen und unendlichen Graphen*, Akademische Verlagsgesellschaft, Leipzig, 1936. (Reprinted by Chelsea, New York, 1950.)

KRATOCHVÍL, J., TUZA, Z., and VOIGT, M., "New Trends in the Theory of Graph Colorings: Choosability and List Coloring," in R. L. Graham, J. Kratochvíl, J. Nešetřil, and F. S. Roberts (eds.), *Contemporary Trends in Discrete Mathematics*, DIMACS Series, 49, American Mathematical Society, Providence, RI, 1999, 183–195.

KREHER, D. L., and STINSON, D. R., *Combinatorial Algorithms: Generation, Enumeration, and Search*, CRC Press, Boca Raton, FL, 1998.

KURATOWSKI, K., "Sur le Problème des Courbes Gauches en Topologie," *Fund. Math.*, *15* (1930), 271–283.

LEHMER, D. H., "The Chromatic Polynomial of a Graph," *Pacific J. Math.*, *118* (1985), 463–469.

LIU, C. L., *Topics in Combinatorial Mathematics*, Mathematical Association of America, Washington, DC, 1972.

LIU, D.-F., "Graph Homomorphisms and the Channel Assignment Problem," Ph.D. thesis, University of South Carolina, 1991.

LOVÁSZ, L., "Normal Hypergraphs and the Perfect Graph Conjecture," *Discrete Math.*, *2* (1972), 253–267. (a)

LOVÁSZ, L., "A Characterization of Perfect Graphs," *J. Comb. Theory B*, *13* (1972), 95–98. (b)

LUKS, E. M., "Isomorphism of Graphs of Bounded Valence Can Be Tested in Polynomial Time," *J. Comput. System Sci.*, *25* (1982), 42–65.

MACKENZIE, D., "Graph Theory Uncovers the Roots of Perfection," *Science*, *297* (2002), 38.

MAHADEV, N. V. R., and ROBERTS, F. S., "Consensus List Colorings of Graphs and Physical Mapping of DNA," in M. Janowitz, F. R. McMorris, B. Mirkin, and F. S. Roberts (eds.), *Bioconsensus*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 61, American Mathematical Society, Providence, RI, 2003, 83–95.

MAKARYCHEV, Y., "A Short Proof of Kuratowski's Graph Planarity Criterion," *J. Graph Theory*, *25* (1997), 129–131.

MANBER, U., *Introduction to Algorithms: A Creative Approach*, Addison-Wesley Longman, Reading, MA, 1989.

MCKEE, T. A., and BEINEKE, L. W., *Graph Theory in Computer Science, Chemistry, and Other Fields*, Pergamon Press, Exeter, UK, 1997.

MOON, J. W., "Various Proofs of Cayley's Formula for Counting Trees," in F. Harary (ed.), *A Seminar on Graph Theory*, Holt, Rinehart and Winston, New York, 1967, 70–78.

OPSUT, R. J., and ROBERTS, F. S., "On the Fleet Maintenance, Mobile Radio Frequency, Task Assignment, and Traffic Phasing Problems," in G. Chartrand, Y. Alavi, D. L. Goldsmith, L. Lesniak-Foster, and D. R. Lick (eds.), *The Theory and Applications of Graphs*, Wiley, New York, 1981, 479–492.

PESCHON, J., and ROSS, D., "New Methods for Evaluating Distribution, Automation, and Control (DAC) Systems Benefits," *SIAM J. Algebraic Discrete Methods*, *3* (1982), 439-452.

232

PONOMARENKO, I. N., "A Polynomial Isomorphism Algorithm for Graphs Not Contractible to $K_{3,g}$ (Russian; English summary)," *Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov. (LOMI)*, *137* (1984), 99–114.

PONOMARENKO, I. N., "Polynomial Time Algorithms for Recognizing and Isomorphism Testing of Cyclic Tournaments," *Acta Appl. Math.*, *29* (1992), 139–160.

PRIM, R. C., "Shortest Connection Networks and Some Generalizations," *Bell Syst. Tech. J.*, *36* (1957), 1389–1401.

RAYCHAUDHURI, A., "Intersection Assignments, T-Coloring, and Powers of Graphs," Ph.D. thesis, Rutgers University, 1985.

READ, R. C., "An Introduction to Chromatic Polynomials," *J Comb. Theory, 4* (1968), 52–71.

REINGOLD, E. M., NIEVERGELT, J., and DEO, N., *Combinatorial Algorithms: Theory and Practice*, Prentice Hall, Englewood Cliffs, NJ, 1977.

ROBERTS, F. S., *Discrete Mathematical Models, with Applications to Social, Biological, and Environmental Problems*, Prentice Hall, Englewood Cliffs, NJ, 1976.

ROBERTS, F. S., *Graph Theory and Its Applications to Problems of Society*, NSF-CBMS Monograph No. 29, SIAM, Philadelphia, 1978.

ROBERTS, F. S., "Indifference and Seriation," *Ann. N.Y. Acad. Sci., 328* (1979), 173–182.

ROBERTS, F. S., "From Garbage to Rainbows: Generalizations of Graph Coloring and their Applications," in Y. Alavi, G. Chartrand, O. R. Oellermann and A. J. Schwenk (eds.), *Graph Theory, Combinatorics, and Applications*, Vol. 2, Wiley, New York, 1991, 1031–1052.

ROBERTSON, N., SANDERS, D. P., SEYMOUR, P. D., and THOMAS, R., "The Four Colour Theorem," *J. Comb. Theory, Series B, 70* (1997), 2–44.

ROUVRAY, D. H., and BALABAN, A. T., "Chemical Applications of Graph Theory," in R. J. Wilson and L. W. Beinecke (eds.), *Applications of Graph Theory*, Academic Press, London, 1979, 177–221.

SCOTT, S. H., "Multiple Node Colourings of Finite Graphs," doctoral dissertation, University of Reading, England, March 1975.

SHIER, D. R., "Testing for Homogeneity using Minimum Spanning Trees," *UMAP J., 3* (1982), 273–283.

SHOR, P. W., "A New Proof of Cayley's Formula for Counting Labeled Trees," *J. Combin. Theory, Ser. A, 71* (1995), 154–158.

STEEL, M. A., "The Complexity of Reconstructing Trees from Qualitative Characters and Subtrees," *J. Classification, 9* (1992), 91–116.

TAKÁCS, L., "On Cayley's Formula for Counting Forests," *J. Combin. Theory, Ser. A, 53* (1990), 321–323.

TESMAN, B. A., "Complete Graph T-Spans," *Congressus Num., 35-A* (1993), 161–173.

TUCKER, A. C., "Perfect Graphs and an Application to Optimizing Municipal Services," *SIAM Rev., 15* (1973), 585–590.

TUCKER, A. C., *Applied Combinatorics*, 2nd ed., Wiley, New York, 1984.

TUTTE, W. T. (alias B. DESCARTES), "Solution to Advanced Problem No. 4526," *Amer. Math. Monthly, 61* (1954), 352.

VAN DEN HEUVEL, J., LEESE, R. A., and SHEPHERD, M. A., "Graph Labeling and Radio Channel Assignment," *J. Graph Theory, 29* (1998), 263–283.

WAGNER, K., "Über Eine Eigenschaft der Ebene Komplexe," *Math. Ann., 114* (1937), 570–590.

233

WANG, D.-I., "The Channel Assignment Problem and Closed Neighborhood Containment Graphs," Ph.D. thesis, Northeastern University, 1985.

WELSH, D., and WHITTLE, G. P., "Arrangements, Channel Assignments, and Associated Polynomials," *Adv. Appl. Math., 23* (1999), 375–406.

WEST, D. B., *Introduction to Graph Theory*, Prentice Hall, Upper Saddle River, NJ, 2001.

WHITNEY, H., "The Coloring of Graphs," *Ann. Math., 33* (1932), 688–718.

WILKINSON, E. M., "Archaeological Seriation and the Traveling Salesman Problem," in F. R. Hodson, *et al.* (eds.), *Mathematics in the Archaeological and Historical Sciences*, Edinburgh University Press, Edinburgh, 1971.

ZYKOV, A. A., "On Some Properties of Linear Complexes (Russian)," *Mat. Sbornik N.S., 24* (1949), 163–188. (English transl.: *Amer. Math. Soc. Transl., 1952*, (1952), 33pp.)

234

第4章 关 系[⊖]

4.1 关系

同日常生活一样，科学领域中经常要弄清楚两个对象、两个项或两个选择之间有什么关系．例如：a大于b，a比b声音高，a是b的兄弟，a比b更令人喜爱，或a和b同样能干等等．本章精确地描述对象之间的关系，特别是二元关系，并指出二元关系的研究与第3章的有向图的研究有着密切的关系．我们特别注意定义所谓的顺序关系，并把这些关系应用于诸如计算机科学、经济学、心理学、生物学和考古学等领域．

4.1.1 二元关系

假设X和Y是集合．X与Y的*笛卡儿积*(cartesian product)，记作$X\times Y$，是所有序对(a, b)的集合，其中a属于X，b属于Y．集合X上的*二元关系*(binary relation)是笛卡儿积$X\times X$的子集，即是序对(a_1, a_2)的集合，其中a_1和a_2都属于X．为了强调基础集合的重要性，我们通常说二元关系(X, R)，而不是只说二元关系R．如果X是集合$\{1, 2, 3, 4\}$，X上的二元关系的例子如下所示：

$$R=\{(1,1),(1,2),(2,1),(3,1),(3,2),(3,3),(3,4)\}, \tag{4.1}$$

$$S=\{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)\}. \tag{4.2}$$

235 回头看3.1.2节，我们看到二元关系的定义方法与有向图完全一样．回想一下，有向图被定义为对(V, A)，其中V是一个集合，A是V中的元素的序对的集合；这时A是集合V上的一个二元关系；即A是笛卡儿积$V\times V$的一个子集的另一种陈述．对应于(4.1)式和(4.2)式定义的二元关系的有向图如图4.1所示．注意，因为本书中所有的有向图都有有限的顶点集合，我们将只讨论有限集合X上的二元关系的有向图．

对于集合X上的二元关系R，我们通常用aRb来表示以下陈述：$(a, b)\in R$或在R的有向图中存在一条从a到b的弧．因此，例如，如果S是(4.2)式的二元关系[⊜]，那么有$1S4$和$2S3$，但没有$3S1$．我们还使用$\sim aRb$表示陈述：(a, b)不在R中，或者在R的有向图中不存在从a到b的弧．

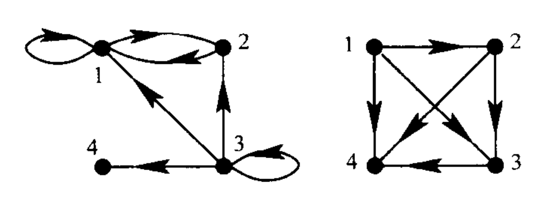


图4.1 二元关系(4.1)式和(4.2)式的有向图表示

正如名字所提示的那样，二元关系表示两个相关的元素，以及某种顺序．在日常生活中经常出现二元关系．例如，如果X是全世界所有人的集合，那么集合

$$F=\{(a,b):a\in X \text{且} b\in X \text{且} a \text{是} b \text{的父亲}\}$$

定义X上的一个二元关系，我们称这种二元关系为“的父亲”．

【例4.1 优先选择】 假设X是你从中选择的任意事物的集合，例如，正餐的食谱、总统候选人的集合或工作候选人的集合，或者是软件包的集合等等．假设

⊖ 本章可以跳过．建议把3.1.2节作为必要知识．另外，可以先跳过本章，后面再返回来学习．本章的概念只在本书的几个小节的若干例子中需要，特别在第8章、第12章和第13章中的部分小节．

⊜ 我们常常使用术语“关系”来表示“二元关系”．更一般地，关系是笛卡儿积$X\times X\times\cdots\times X$的子集．

$$P = \{(a,b) \in X \times X: 喜欢 b 胜过喜欢 a\}.$$

那么，P 可以称为集合 X 上你的**严格优先选择**(strict preference)关系. 严格优先选择不同于**弱优先选择**(weak preference)：前者意指“与……相比更”，而后者意指“至少与……一样”. 通常我们将优先选择看作或严格或弱. 关系(X, P)在经济学、政治学、心理学和其他领域得到广泛的研究. 作为一个具体的例子，假设你正在考虑在你的可选择的休假目的地之间作出优先选择，你的可能目的地集合是 X={旧金山，洛杉矶，纽约，波士顿，迈阿密，亚特兰大，菲尼克斯}，你的严格选择关系是由集合 P 给出的：P={(波士顿，亚特兰大)，(波士顿，菲尼克斯)，(波士顿，旧金山)，(波士顿，迈阿密)，(波士顿，洛杉矶)，(纽约，亚特兰大)，(纽约，菲尼克斯)，(纽约，旧金山)，(纽约，迈阿密)，(纽约，洛杉矶)，(旧金山，洛杉矶)，(亚特兰大，旧金山)，(亚特兰大，洛杉矶)，(迈阿密，旧金山)，(迈阿密，洛杉矶)，(迈阿密，亚特兰大)，(迈阿密，菲尼克斯)，(菲尼克斯，洛杉矶)，(菲尼克斯，旧金山)}. 因此，与亚特兰大相比较，你严格倾向于去迈阿密. 对应于该(X, P)的有向图如图 4.2 所示. ■

236

图 4.2 例 4.1 的数据的优先选择有向图

【例 4.2 心理物理量表法】 刺激的物理性质与心理性质之间的关系的研究称为**心理物理学**(psychophysics). 例如，在心理物理学中，我们尝试着将响度、亮度或甜度的心理反应与声音、光亮或食物相联系(关于从数学角度介绍心理物理学请参见 Falmagne[1985]). 我们通常先进行比较. 例如，如果 X 是诸如来自距我们不同距离的不同飞机的声音集合，我们也许说“一架飞机的声音比另外一架的声音大”. 如果 aLb 意指“a 的声音大于 b 的声音”，那么(X, L)是一个二元关系. 例如，设 a 是距离我们 2000 英尺[⊖]处的一架波音 747，a'是距离我们 3000 英尺处的一架波音 747，b 是距离我们 2000 英尺处的一架波音 757，b'是距离我们 3000 英尺处的一架波音 757，c 是距离我们 2000 英尺处的一架波音 767，c'是距离我们 3000 英尺处的一架波音 767. 假设

$$L = \{(a,a'),(a,b),(a,b'),(a,c),(a,c'),(a',b),(a',b'),(a',c'),(b,b'),(b,c'),(b',c')(c,c')\}.$$

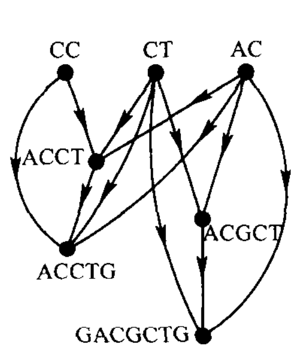


图 4.3 心理物理量表法的“比……声音大”的有向图. 该有向图对应于例 4.2 的关系(X, L)

那么，例如，a 的声音大于 c 的声音. 对应于(X, L)的有向图如图 4.3 所示. ■

237

【例 4.3 子串问题】 在生物科学和计算机科学中，我们研究某个字母表上的符号串. 我们通常对一个串是否作为另一个串的连续子串出现感兴趣. 这在分子生物学中非常重要，在这一领域，我们在诸如 DNA 或 RNA 这样的大分子序列中寻找被定义为小的连续子串的“模式”. 我们将在例 11.2 和 11.6.5 节再次研究相关的思想.

假设 X 是一个串的集合. 我们以 aSb 表示串 a 作为串 b 的连续子串出现. 这定义一个二元关系(X, S). 作为具体例子，设

$$X = \{CC, CT, AC, ACCT, ACCTG, ACGCT, GACGCTG\}.$$

于是，我们有(CC, ACCT)∈S，(ACCT, ACCTG)∈S，等等. 对应

图 4.4 对应于例 4.3 的关系(X, S)的有向图

⊖ 英尺的单位符号为 ft，1ft=0.3048m. ——编辑注

的有向图如图 4.4 所示.

二元关系(X, S)在计算机科学中起着重要的作用. 经常出现的问题是，给定一个串的固定集合，要求确定给定的串是否是这一集合中的任意串的连续子串. 称为后缀树的数据结构在解决这一问题中起到重要作用. 对这一问题的讨论参见 Gusfield[1997]. ■

【例 4.4 搜索引擎】 诸如 Google 等的网络搜索引擎使用咨询和网页间的相关性度量. 我们开始于一个网页集合 X. 当然，由于因特网在不断地发展和变化，集合 X 也在变化，但是我们假设在任意给定时刻它是固定的. 搜索引擎带来的挑战之一是在任意给定时刻寻找集合 X. 设 q 是一个咨询，x 是一个网页，且 $r(x, q)$是网页 x 与咨询 q 间的相关性度量. 搜索引擎的另一个挑战是确定如何度量 $r(x, q)$. 我们说 x 的等级高于 y，如果$r(x, q)>r(y, q)$，在这种情况下，我们记为 xRy. 例如，考虑表 4.1 中 $r(x, q)$的值，因此有

表 4.1　网页 x 及其对于咨询 q 的相关性

x	a	b	c	d	e	f
$r(x, q)$	7	8	5	7	8	4

$$R=\{(b,a),(b,c),(b,d),(b,f),(e,a),(e,c),(e,d),(e,f),(a,c),(a,f),(d,c),(d,f),(c,f)\}.$$

其对应的有向图如图 4.5 所示. ■

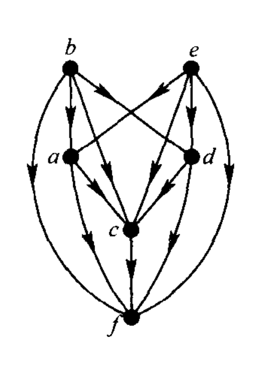


图 4.5　对应于例 4.4 的关系(X, R)的有向图

【例 4.5 考古学中的定序】 很多应用中的一个普遍问题涉及把项或个体或事件以基于有关它们的某些信息按自然顺序排列起来. 例如，在考古学中，在不同的古迹挖掘中会发现若干种类的陶器或其他人工制品. 我们希望以对应于这些人工制品所存在的历史年代把它们排序. 例如，我们知道在时间上人工制品 a 超前于 b. 对于这些制品我们能重排一个顺序吗? 称为*序列年代*(sequence dating)、*定序*(sequencing)或*排序*(seriation)的问题要追溯到 Flinders Petrie[1899, 1901]的工作. 在 Kendall[1963, 1969a, b]中可以找到其他关于序列年代的数学讨论. 作为一个具体例子，假设 X 是由五个类型a, b, c, d, e 的陶器组成的，我们知道 a 超前于 c、d，且 e、b 超前于 d、e，且 c 超前于 e. 那么如果 xQy 意指 x 超前于 y，那么我们有

$$Q=\{(a,c),(a,d),(a,e),(b,d),(b,e),(c,e)\}.$$

对应于(X, Q)的有向图如图 4.6 所示. ■

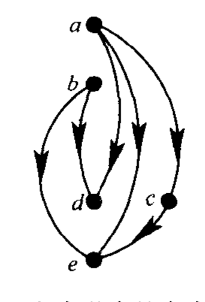


图 4.6　考古学中的定序的领先有向图. 有向图对应于例 4.5 的关系(X, Q)

4.1.2　关系的性质/有向图中的模式

很多自然的关系都有若干共同性质. 本节讨论其中的一些性质，以及当考虑带有这些性质的关系的有向图时，这些性质的表示. 我们用表 4.2 来概括这些性质.

表 4.2　关系的性质

二元关系(X, R)	假　设
自反的	对于所有 $a\in X$，有 aRa
非自反的	不是自反的
反自反的	对于所有 $a\in X$，有$\sim aRa$
对称的	对于所有 a，$b\in X$，有 $aRb\Rightarrow bRa$
非对称的	不是对称的
不对称的	对于所有 a，$b\in X$，$aRb\Rightarrow\sim bRa$
反对称的	对于所有 a，$b\in X$，aRb 且 $bRa\Rightarrow a=b$

（续）

二元关系(X，R)	假　设
传递的	对于所有 a，b，$c\in X$，aRb 且 $bRc\Rightarrow aRc$
非传递的	不是传递的
负传递的	对于所有 a，b，$c\in X$，$\sim aRb$ 且 $\sim bRc\Rightarrow\sim aRc$ 或对于所有 a，b，$c\in X$，$aRc\Rightarrow aRb$ 或 bRc
强完全的	对于所有 a，$b\in X$，aRb 或 bRa
完全的	对于所有 $a\neq b\in X$，aRb 或 bRa

二元关系(X, R)是自反的(reflexive)，如果对于所有 $a\in X$，有 aRa. 因此，例如，如果 X 是一个数的集合，且 R 是 X 上的关系"相等"，那么(X, R)是自反的，因为一个数总是等于其本身. 然而，如果 $X=\{1, 2, 3, 4\}$，(4.1)式的关系 R 不是自反的，因为 $2R2$(和 $4R4$)不成立. 在这一情况下，这一二元关系被称之为非自反的(nonreflexive)，其意思就是"不是自反的". 还有如果 $X=\{1, 2, 3, 4\}$，(4.2)式的关系是非自反的. 因为$\sim 1S1$(以及$\sim 2S2$，$\sim 3S3$，$\sim 4S4$). 当二元关系的非自反性有如这一关系时，则它被称为是反自反的(irreflexive). 也就是说，(X, R)是反自反的，如果对于所有的 $a\in X$，有$\sim aRa$. 在这一意义下，人的集合上的关系"的父亲"是反自反的. 例 4.1、4.2、4.3、4.4 和 4.5 的关系(X, P)、(X, L)、(X, S)、(X, R)和(X, Q)也都是反自反的. 从有向图的角度来说这些意味着什么呢？如果二元关系是自反的，那么它的有向图在每一个顶点处都有一个环. 在非自反关系中，至少在一个顶点处没有环；在反自反关系中，(每个顶点都)没有环存在. 因此，很容易从一个关系的有向图中发现这三个性质的存在与否. 考虑图 4.7 的有向图. 我们很快就确定图 4.7a 是反自反和非自反的，图 4.7b 仅是非自反的，而图 4.7c 是自反的.

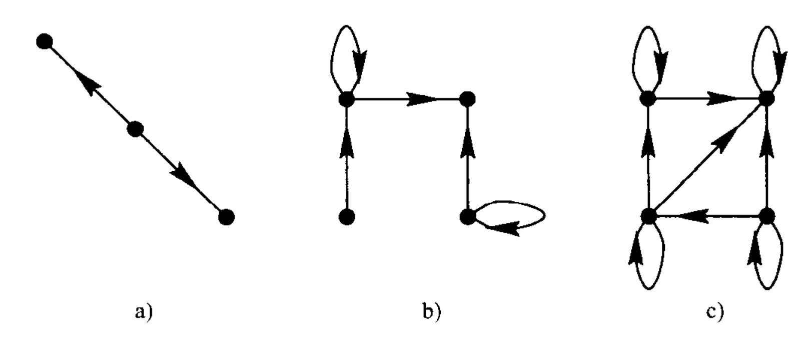


图 4.7　有向图的例子

在 3.1.2 节中，对于一个有向图，我们通过检查其中是否只要存在从 v 到 u 的弧就存在从 u 到 v 的弧来定义图. 对于肯定的情况，我们说这一有向图是一个图，且使用一条称为边的非有向线取代顶点间的每一对弧. 只要存在从 v 到 u 的弧，则存在一条从 u 到 v 的弧的条件正好是下面性质的条件. 二元关系(X, R)称为对称的(symmetric)，如果对于所有 a，$b\in X$，有

$$aRb\Rightarrow bRa.$$

即(X, R)是对称的，那么只要$(a, b)\in R$ 就有$(b, a)\in R$. 所以，根据上面的讨论，任意一个图都表示一个对称的二元关系(注意，对称的有向图可能存在环也可能不存在环). 一个数的集合上的二元关系"相等"是对称的. 美国所有男性的集合上的关系"的兄弟"是对称的. 然而，美国所有人的集合上的关系"的兄弟"则不是对称的，因为如果 a 是 b 兄弟，不一定满足 b 是 a 的兄弟(为什么). 这也表明在定义关系和研究它的性质时，说明其基础集合的重要性.

其他非对称(nonsymmetric)关系(不是对称的关系)的例子分别是世界所有人的集合上的关系"的父亲"，以及例 4.1、4.2、4.3、4.4 和 4.5 中的关系(X, P)、(X, L)、(X, S)、(X, R)、

(X, Q)等等. 这五个关系的非对称性都非常高. 事实上，它们是*不对称的*(asymmetric)，因为它们满足下面的规则：

$$aRb \Rightarrow \sim bRa.$$

其他不对称关系包括实数集合上的关系"大于"，集合的任意集合上的"严格包含于"$\subsetneqq$，(4.2)式的集合 $X=\{1, 2, 3, 4\}$上的关系 S 等等. 刻画一个不对称关系的有向图具有什么样的性质呢？一种解释是一个不对称关系的有向图没有环，且对于所有顶点 u 和 v，有

$$d(u,v)+d(v,u) \neq 2. \tag{4.3}$$

240~241

一些关系(X, R)不是完全不对称的，但在下面的意义下几乎是不对称的：允许有环，但对于顶点 $u \neq v$，公式(4.3)式成立. 我们称(X, R)是*反对称的*(antisymmetric)，如果对于所有 a，$b \in X$，有

$$aRb \text{ 且 } bRa \Rightarrow a=b.$$

所以，反对称有向图类似于允许有环的不对称有向图. 在很多例子中，相对于不对称性，反对称意味着允许有"相等的元素". 例如，一个实数集合上的关系"大于或等于"$\geqslant$、集合的任意集合上的关系"包含于"$\subseteqq$、人(其中没有两个人有同样的高度)的任意集合上的"至少身高相同"是三个反对称关系的例子. 容易证明，每一个不对称的二元关系是反对称的，但是逆叙不成立(练习 15).

关系(X, R)称为*传递的*(transitive)，如果对于所有的 a，b，$c \in X$，当 aRb 且 bRc 时有 aRc. 即(X, R)是传递的，如果对于所有 a，b，$c \in X$，

$$aRb \text{ 且 } bRc \Rightarrow aRc.$$

传递关系的例子有实数集合上的关系$=$和$>$，陈述集合上的关系"蕴涵"，以及关系(X, S)，其中 $X=\{1, 2, 3, 4\}$及 S 由(4.2)式给出. 例 4.1、4.2、4.3、4.4 和 4.5 的关系(X, P)、(X, L)、(X, S)、(X, R)、(X, Q)都是传递关系. 对于有精确定义的(X, S)和(X, R)这是显然的. 对于其他三个例子似乎也是合理的. 因此认为在可选择休假目的地集合上的严格优先选择关系总是传递的也是合理的，因为，如果 a 优先于 b 且 b 优先于 c，那么应该希望 a 优先于 c. 类似地，认为飞机集合上的关系"声音大于"总是传递的也是合理的；同样，人工制品集合上的关系"超前于"也应该总是传递的. 同上面的三个例子的情况一样，这只要用实验数据加以证实. 对于现实应用中出现的严格优先选择关系，有时候我们发现传递性被破坏. 如果 $X=\{1, 2, 3, 4\}$且 R 由(4.1)式给出，那么关系(X, R)是*非传递的*(nontransitive)，即不是传递的，因为有 $2R1$ 和 $1R2$，但是$\sim 2R2$. 另一个不是传递的关系是世界所有人的集合上的关系"的父亲". 然而，如何从有向图得知与它相关的二元关系是否是传递的呢？传递性意味着如果存在一条从顶点 u 到顶点 v 的弧和一条从顶点 v 到顶点 w 的弧，那么必将存在一条从顶点 u 到顶点 w 的弧. 我们也可以通过对有向图的距离函数的限制定义传递性(参见练习 28).

我们的下一个性质类似于传递性，但是在反面意义下的传递性. 二元关系(X, R)是*负传递的*(negatively transitive)，如果对于所有 a，b，$c \in X$，$\sim aRb$ 且$\sim bRc$ 意味着$\sim aRc$. 二元关系(X, R)是负传递的，如果定义在 X 上的关系"不在 R 中"是传递的. 作为例子，实数集合上的关系 $R=$"大于"是负传递的，因为"不在 R 中"是关系"不大于"，或"小于或等于"，而这一关系当然是传递的. 容易证明，如果 $X=\{1, 2, 3, 4\}$，(4.2)式的关系 S 是负传递的. 例 4.3 和例 4.4 的关系(X, S)和(X, R)也是负传递的. 类似地，可选集合上的严格优先选择关系、飞机集合上的关系"声音大于"、人工制品集合上的超前关系有可能都是负传递关系(当研究经济学、政治学、心理学、考古学时，"可能"这样的词汇通常是可以期待的最佳词汇). 证实负传递性可能很混乱. 通常验证它的逆否(因此，是等价的)条件会容易得多：对于所有 a，b，$c \in X$，如果 aRc，则 aRb 或 bRc. 使用这一概念，我们很容易看到关系"大于"是负传递的，因为如果 $a>c$ 那么对于所有 b，则有 $a>b$ 或者 $b>c$. 类似地，你还可以看到关系"包含于"不是负传递的，因为如果 a 包含于

242

c，则很可能存在一个b使得a不包含于b，且b不包含于c(参见图4.8中的例子). 世界所有人的集合上的关系"的父亲"不是负传递的，关系(X, R)也不是负传递的，其中$X=\{1, 2, 3, 4\}$，且R是由(4.1)式给出的. 对于后者，注意$2R1$，但是没有$2R4$和$4R1$.

验证有向图的负传递性可能不是一项简单的工作. 在最坏情况下，对于每条弧(u, v)，必须对每一个不等于u和v的顶点w搜索两条弧(u, w)和(w, v). 因此，验证负传递性将花费$2a(n-2)$次搜索，其中n和a分别是顶点集合的大小和这一有向图的弧集合的大小.

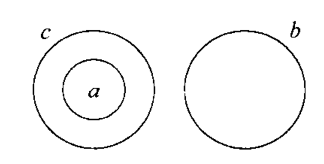


图4.8 关系"包含于"不一定是负传递的

我们说二元关系(X, R)是**强完全的**(strongly complete)，如果对于所有$a, b\in X$，有aRb或bRa. 因此，实数集合上的关系"大于或等于"$\geqslant$是强完全的. 然而，集族上的"严格包含"则可能不是强完全的，因为，可能存在两个互不严格包含的集合. 类似地，世界所有人的集合上的关系"的父亲"不是强完全的(为什么). 例4.3的子串关系也不是强完全的. 强完全性的有向图测试要涉及其基础图(参见练习6).

注意，数的集合上的关系"大于"不是强完全的，因为如果$a=b$，那么$\sim a>b$且$\sim b>a$. 在下面的意义下这一关系几乎是强完全的：对于所有$a\neq b$，有aRb或bRa. 满足这一条件的二元关系称为**完全的**(complete). $X=\{1, 2, 3, 4\}$上由等式(4.1)定义的关系R既不是完全的也不是强完全的(为什么)，但是X上的由等式(4.2)定义的关系S是完全的，但不是强完全的. 例4.4的关系(X, R)既不是完全的也不是强完全的. (对于强完全性的有向图表示参见练习6.)

本节练习

1. (a)考虑正整数集合上的二元关系"a整除b". 这一关系有下列哪些性质：自反、反自反、对称、不对称、反对称、传递、负传递?
   (b)对于人的集合上的关系"的伯父"，重复(a).
   (c)对于老鼠的集合上的关系"有相同体重"，重复(a).
   (d)对于对象的集合上的关系"感觉比……更光滑"，重复(a).
   (e)对于人的集合上的关系"钦佩"，重复(a)
   (f)对于人的集合上的关系"有相同血型"，重复(a).
   (g)对于汽车的集合上的关系"成本更高"，重复(a).
   (h)对于关系(X, R)，其中X是所有位串的集合，且aRb意味a的某个真后缀是b的一个真前缀(串$b_1b_2\cdots b_n$的**真后缀**(proper suffix)是形如$b_ib_{i+1}\cdots b_n$的串，$n\geqslant i>1$；真前缀的定义类似)，重复(a).

243

2. 证明：世界所有人的集合上的二元关系"的兄弟"不是对称的.
3. 证明：世界所有人的集合上的二元关系"的父亲"不是强完全的.
4. 寻找可以用于描述等式(4.2)的二元关系的"日常生活"例子.
5. (Stanat和McAllister[1977])给我们一个由集合Y组成的文件库，并开发了一个"描述符"(如关键字)的集合Z来描述这些文件. 设$X=Y\cup Z$，且设aRb成立，如果描述符b适应于文件a. 文件检索系统使用关系(X, R)来为用户寻找相关文件. 关系(X, R)满足练习1(a)中的哪些性质?
6. (a)描述：当且仅当有向图D强完全时，其基础图必须满足的检测.
   (b)如果使用完全取代强完全，那么(a)的答案将如何变化?
7. 如果(X, R)是一个二元关系，R的**逆**(converse)是X上由下式定义的关系R^{-1}：

$$aR^{-1}b \text{ 当且仅当 } bRa.$$

   (a)描述与R相对照的R^{-1}有向图.
   (b)确定所有瑞典人的集合上的二元关系"的伯父"的逆.
8. 如果(X, R)是一个二元关系，R的**补**(complement)是X上由下式定义的关系R^c：

$$aR^cb \text{ 当且仅当 } \sim aRb.$$

(a)描述与 R 相对照的 R^c 的有向图.

(b)证明或给出下面陈述的反例：如果 R 是对称的，那么 R^c 是对称的.

(c)确定所有斯里兰卡人的集合上的二元关系"的父亲"的补.

9. 如果(X, R)和(X, S)是二元关系，X 上的交(intersection)关系 $R\cap S$ 由下式定义：

244

$$R\cap S=\{(a,b): aRb \text{ 且 } aSb\}.$$

(a)当 X=所有爱尔兰人的集合，R="年龄大"，S="的父亲"时，确定$(X, R\cap S)$.

(b)当 $X=\{1, 2, 3, 4\}$且 R 和 S 分别是(4.1)式和(4.2)式给出的关系时，画出$(X, R\cap S)$的有向图.

10. 假设(X, R)和(X, S)是二元关系. 对于练习 1(a)所列出的每一个性质，假设(X, R)和(X, S)都有所述的性质. 这时，或者证明$(X, R\cap S)$也有该性质，或者给出例子证明它可能不具有该性质.

11. 对于关系(X, R^{-1})重复练习 10.

12. 对于关系(X, R^c)重复练习 10.

13. $(X, \varnothing)$满足练习 1(a)中的哪些性质？

14. $(X, X\times X)$满足练习 1(a)中的哪些性质？

15. (a)证明：每一种不对称关系是反对称的.

 (b)证明：(a)的逆不成立.

16. 证明：不对称二元关系是反自反的.

17. 证明：二元关系是不对称且传递的，当且仅当它是反自反、反对称且传递的.

18. (a)证明：一个二元关系不可能既是对称的又是不对称的.

 (b)证明：一个二元关系不可能既是对称的又是反对称的.

19. 证明：存在有下列性质的二元关系：

 (a)传递但不是负传递的.

 (b)负传递但不是传递的.

 (c)既不是负传递的也不是传递的.

 (d)既是负传递的又是传递的.

20. 二元关系(X, R)是等价关系(equivalence relation)，如果它是自反、对称且传递的. 下列二元关系中哪些是等价关系？

 (a)$(Re, =)$，其中 Re 是实数集合.

 (b)$(Re, \geqslant)$.

 (c)$(Re, >)$.

 (d)X=人的集合，aRb 当且仅当 a 和 b 有相同体重.

 (e)$X=\{0, 1, 2, \cdots, 22\}$，$aRb$ 当且仅当 $a\equiv b(\bmod\ 5)$.

 (f)X 是实数的所有有限集合的集合且 aRb 当且仅当 $a\cap b\neq\varnothing$.

 (g)$X=\{(1, 2), (2, 3), (3, 8)\}$且 $R=\{\langle(1, 1), (1, 1)\rangle, \langle(2, 3), (2, 3)\rangle, \langle(3, 8), (3, 8)\rangle, \langle(1, 1), (2, 3)\rangle, \langle(2, 3), (1, 1)\rangle\}$.

21. 如果(X, R)和(X, S)是等价关系(参见练习 20)，那么

 (a)$(X, R\cap S)$是等价关系吗？其中 $R\cap S=\{(a, b): aRb \text{ 且 } aSb\}$.

245

 (b)$(X, R\cup S)$是等价关系吗？其中 $R\cup S=\{(a, b): aRb \text{ 或 } aSb\}$.

 (c)$(X, R/S)$是等价关系吗？其中 $R/S=\{(a, b) \mid \text{对于某个 } c\in X, \text{有 } aRc \text{ 且 } cSb\}$.

22. 为了说明等价关系所需的所有性质，给出有下列性质的二元关系例子：

 (a)自反、对称但不是传递的.

 (b)自反、传递但不是对称的.

 (c)对称、传递但不是自反的.

23. 如果(X, R)是一个等价关系(参见练习 20)，设 $C(a)=\{b\in X \mid aRb\}$. 这一集合称为包含 a 的等价类(equivalence class containing a). 例如，如果 $X=\{1, 2, 3, 4, 5\}$且 $R=\{(1, 1), (1, 2), (2, 1),$

(2, 2), (3, 3), (4, 4), (5, 5)}，那么(X, R)是一个等价关系．等价类有$C(1)=C(2)=\{1, 2\}$，$C(3)=\{3\}$，$C(4)=\{4\}$，$C(5)=\{5\}$.

(a)寻找练习 20(e)中的等价关系的所有等价类.

(b)证明：两个等价类$C(a)$和$C(b)$或者不相交或者相等.

(c)给出一个有三个不同等价类的等价关系的例子.

(d)给出有两个不同的等价类，其中一个有三个元素，另一个有两个元素的等价关系的例子.

24. 假设$X=Re$，且有

$$aRb \text{ 当且仅当 } a > b+1.$$

对于关系(X, R)，练习 1(a)的哪些性质成立？

25. 考虑二元关系(X, S)，其中$X=Re$，且有

$$aSb \text{ 当且仅当 } |a-b| \leqslant 1.$$

这一关系与练习 24 的二元关系(X, R)有密切的联系．对于(X, S)，练习 1(a)中的哪些性质成立？

26. 如果(X, R)是一个二元关系，R的对称补(symmetric complement)是X上由下式定义的二元关系S：

$$aSb \text{ 当且仅当}(\sim aRb \text{ 且 } \sim bRa).$$

注意，如果R是严格优先选择关系，那么S是中立关系．对于两个选择你保持中立，当且仅当你对其任意一个都不更加喜欢.

(a)证明：对称补总是对称的.

(b)证明：如果(X, R)是负传递的，那么对称补是传递的.

(c)证明：(b)的逆不成立.

(d)如果$X=Re$且R如练习 25 所定义，寻找一个描述R的对称补的不等式.

(e)确定下列关系的对称补：

i. $(Re, >)$.

ii. $(Re, =)$.

iii. (N, R)，其中N是正整数的集合，且xRy意指x不整除y.

246

27. 计算有n个元素的集合X上的二元关系的数量.

28. 给定有向图D，证明D是传递的，当且仅当只要v从u是可达的则$d(u, v)\neq 2$.

4.2 顺序关系及其变形

本节研究称为顺序关系的特殊二元关系和它们的变形.

4.2.1 定义顺序关系的概念

【例 4.6 效用函数】 假设(X, R)是例 4.1 的严格优先选择关系．在经济学或心理学中，有时候我们用一个数值来反映优先选择，使得指定给一个对象的值越大，就越优先选择这一对象．因此，我们会问我们是否能够给X中的每一个选项a指定一个值$f(a)$，使得与b相比严格优先选择a当且仅当$f(a)>f(b)$：

$$aPb \Leftrightarrow f(a) > f(b). \tag{4.4}$$

如果能够做到这一点，那么有时候f被称为*效用函数*(utility function，*序数效用函数*(ordinal utility function))．在决策应用中效用函数非常有用，因为它给出单一数值，我们基于这一数值进行选择，而且它还为各选项提供一个*顺序*．我们通常选择能够使我们的效用(或“期望”效用)最大的一系列行动．在例 4.1 中，可以找到满足(4.4)式的一个效用函数．其中一个例子是

$$\begin{gathered} f(\text{波士顿}) = 5, f(\text{纽约}) = 5, f(\text{迈阿密}) = 4, \\ f(\text{亚特兰大}) = 2, f(\text{菲尼克斯}) = 2, f(\text{旧金山}) = 1, \\ f(\text{洛杉矶}) = 0. \end{gathered} \tag{4.5}$$

这给出 X 上的一个顺序：波士顿和纽约均为首位，其次是迈阿密，然后是亚特兰大和菲尼克斯，其后是旧金山，最后是洛杉矶．关系 (X, P) 既是传递的又是反对称的(事实上，它是不对称的)．事实上，存在满足(4.4)式函数的任意二元关系 (X, P) 都是传递且反对称的(甚至是不对称的)．为了弄明白为什么，注意，如果 aPb 且 bPc，那么(4.4)式意味 $f(a)>f(b)$ 且 $f(b)>f(c)$．因此，有 $f(a)>f(c)$，根据(4.4)式，有 aPc．反对称性的证明类似．

效用的概念至少要追溯到 18 世纪．对这一概念的大部分原始兴趣要追溯到 Jeremy Bentham[1789]．根据 Bentham："效用的意思是任意对象自身的性质，它有助于产生效益、优势、愉快、满意或幸福……"Bentham 公式化度量效用的过程，因为他认为社会应该争取"最大的值带来最大的满意"，即最大化效用．如何度量效用的问题是一个复杂的问题，而且有很多研究这一问题的文章．对此的讨论可以参见 Barberà，Hammond 和 Seide[2004]、Fishburn [1970b]、Keeney 和 Raiffa[1993]、Luce[2000]、Robert[1979b]．效用有很多应用．它们有助于商业决策，诸如何时买或卖，或者买进哪种计算机系统，它也有助于个人决策，诸如生活在什么地方，接受哪份工作邀请等，它还有助于国家政策的选择，诸如新环境管理或国土安全主动性的投资等等． ■

247

满足传递性和反对称性的二元关系 (X, R) 称为*顺序关系*(order relation)，而且我们说 X 由 R 定序．因此，例如，集合族上的"包含于"是一个顺序关系，"严格包含于"也是一个顺序关系，世界所有人的集合上的关系"的后代"，实数集合上的 $\geqslant$ 都是顺序关系．注意，在图 4.2 的有向图中，每一条弧是头向下画出的．如果 (X, R) 是一个传递关系，且它对应的有向图有这种画法，环除外，那么它是反对称的，而且它是一个顺序关系．注意，根据传递性，只要存在从 a 到 b 的路径且其中每一个弧头向下[⊖]，那么 (a, b) 在关系中．

本节定义各种顺序关系，以及与顺序关系密切相关的关系．我们在表 4.3 中概括了这些定义．定义一类顺序关系的性质不是这类关系仅有的性质．然而，在数学里，我们总是尝试着使用最小的性质集合做一个定义．

表 4.3 顺序关系和它们的变形

定义性质	关系类型						
	顺序关系	弱序	严格弱序	线性序	严格线性序	偏序	严格偏序
自反的						√	
对称的							
传递的	√	√		√	√	√	√
不对称的			√		√		√
反对称的	√			√		√	
负传递的			√				
强完全的		√		√			
完全的					√		

注：给定类型的关系可能满足比指出的性质更多的性质．我们只指出其定义的性质．

图 4.9 给出几个顺序关系的不同例子．注意，即使存在一个上升的弧，图 4.9 中的有向图 c 也是一个顺序关系．存在一个使所有弧都是头向下的画法(参见练习 23)．事实上，不难证明下面的定理．

248

定理 4.1 *一个传递的二元关系 (X, R) 是一个顺序关系，当且仅当可以画出它的有向图，*

⊖ 到目前为止，图或有向图中的顶点的位置不是很重要，只有邻接是重要的．现在，我们要考虑顶点的位置．实际上，前面我们介绍根树时，我们已对这一思想有所了解，但是在那里没有强调顶点的位置确定．

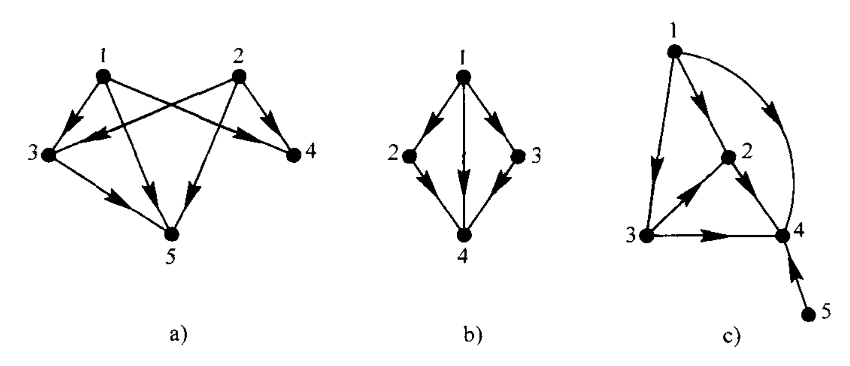


图 4.9 顺序关系的例子

使得所有弧(环除外)都头向下[⊖].

这表明分别对应于图 4.2、图 4.3、图 4.4、图 4.5 和图 4.6 中的有向图的关系(X, P)，(X, L)，(X, S)，(X, R)和(X, Q)都是顺序关系.

顺序关系的两个定义性质有下面定理所示的有趣结论.

定理 4.2 顺序关系的有向图没有循环(环除外).

证明 假设(X, R)是一个顺序关系且$C=a_1, a_2, \cdots, a_j, a_1$是对应的有向图中的一个循环. 因此，$a_1 \neq a_2$[⊜]. 因为$(X, R)$是传递且反对称的，根据$C$的定义，我们知道$(a_1, a_2)$，$(a_2, a_3)$，…，$(a_j, a_1)$是有向图中的弧. 使用弧$(a_2, a_3)$和$(a_3, a_4)$，运用传递性可证$(a_2, a_4) \in R$. 于是，因为$(a_2, a_4) \in R$且$(a_4, a_5) \in R$，传递性暗示着$(a_2, a_5) \in R$. 持续这一过程，我们得出结论$(a_2, a_j) \in R$. 这一结论再加上$(a_j, a_1) \in R$，有$(a_2, a_1) \in R$. 因此，$(a_1, a_2)$和$(a_2, a_1)$是这一有向图的弧，这与该有向图的反对称性相矛盾，因为$a_1 \neq a_2$. ■

可以发现满足(4.4)式的数值表示的二元关系有更多的性质，而不仅仅是传递性和反对称性. 例如，它是负传递的，这给出它的另外一个性质的一个例子(为什么). 在后面的各小节中，我们将通过增加二元关系需要满足的性质来定义更强类型的顺序关系.

除了传递性和反对称性之外，我们将介绍的很多顺序关系或者是自反的或者是反自反的. 这两种可能性通常基于本书所讨论的问题. 如果我们假设有自反性，那么我们称我们的顺序关系(X, R)是一个*偏序*(partial order)或称(X, R)是X的*偏序集合*(partially ordered set)或*偏序集*(poset)[⊜]. 图 4.10a 给出一个偏序. 如果假设反自反性，那么我们将使用形容词"严格"；(X, R)是*严格偏序*(strict partial order)，如果它是反自反、反对称、传递的. 图 4.9a、b、c、图 4.2、图 4.3、图 4.4、图 4.5 和图 4.6 分别给出严格偏序. 有时候，更简洁地把严格偏序定义为不对称且传递的二元关系. 这是因为我们有下面的定理.

249

定理 4.3 二元关系是反自反、传递、反对称的当且仅当它是传递且不对称的.

证明 假设(X, R)是反自反、传递、反对称的. 假设aRb且bRa. 于是根据反对称性，$a=b$. 但是，根据反自反性，aRa为假. 因此，aRb且bRa不成立，(X, R)是不对称的.

⊖ 尽管我们没有明确地加以说明，但是这一定理假设X是有限的.

⊜ 此处不够严密. 应为：假设(X, R)是一个顺序关系且$C=a_1, a_2, \cdots, a_j, a_1$是对应的有向图中的一个包含不是环的弧的循环. 因为如果$a_1=a_2$，那么$a_1, a_3, \cdots, a_j, a_1$满足假设，因此，不妨设$a_1 \neq a_2$. 也可改为：假设$(X, R)$是一个顺序关系且$C=a_1, a_2, \cdots, a_j, a_1$是对应的有向图中最短的不是环的循环. ——译者注

⊜ 有些作者混用"有序集合"和"偏序集合". 我们将加以区分，以供非自反顺序关系之用.

图 4.10 一个偏序的有向图 a 及其相关的图表示 b

反之，假设(X, R)是传递且不对称的．那么 aRb 和 bRa 不都成立，所以(X, R)是反对称的．另外，不对称性蕴涵反自反性，因为 aRa 蕴涵 aRa 且 aRa，根据不对称性，这是不可能的．

■

4.2.2 顺序关系的图表示

考虑集合 X 上的顺序关系 R．因为反对称性和传递性是定义性质，这些性质可用于简化这一关系的有向图．我们使用图 4.10 中的有向图 a 进行说明．因为所有弧(环除外)都指向下方，那么为什么不除去所有弧的箭头，从而把它们变成边呢？另外，这一有向图是自反的，所以没有必要在每一个顶点处画出一个环．这两个变化把这一有向图转换成无环图．接下来，考虑从 3 到 6，从 6 到 7 和从 3 到 7 的边．因为已知传递性成立，所以从 3 到 7 的边是不必要的．一般地，我们可以除去传递性蕴涵的所有边．这将在很大程度上简化顺序关系的有向图(图)．除去环、箭头和传递性蕴涵的弧后生成的图称为这一顺序关系的**图表示**(diagram)(**顺序图表示**(order diagram)或 Hasse 图表示(Hasse diagram))．图 4.10b 给出图 4.10a 的顺序关系的图表示．在一个图表示中，aRb，如果存在一个从 a 到 b 的下降链．例如，在图 4.10b 中，$1R5$，$3R7$，但是$\sim 3R5$．我们也可以对反自反的顺序关系的有向图实施同样的简化．对反自反顺序关系的有向图实施同样简化所得的图也称为图表示．与图 4.9 的有向图 a、b、c 相关联的图表示如图 4.11 所示．

250

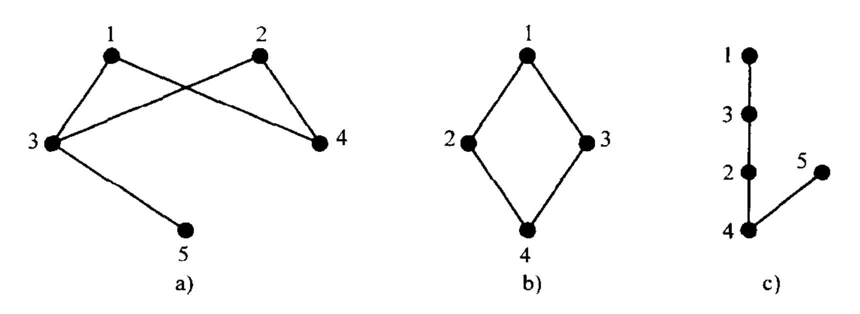


图 4.11 图 4.9 中的顺序关系 a、b 和 c 的图表示

类似地，也可以把任意没有水平边的无环图(即顺序关系)，简化成图表示．图中传递性蕴涵的边(弧)可以除去．图表示不包含环，所以我们将不可能从图表示得知这一关系的自反性．只有通过这一关系的定义或给出的相关内容才能确定自反性．

考虑集合 X 上的顺序关系 R．我们说 x **覆盖**(cover)y 或 xKy，如果 xRy 且不存在 z，使得

xRz 且 zRy. 二元关系(X, K)称为(X, R)的**覆盖关系**(cover relation). 我们可以如下定义与 R 相关的**覆盖图** G_K: $(a)V(G_K)$是集合 X; $(b)\{x, y\}\in E(G_K)$, 如果 x 覆盖 y. 因此, 与一个顺序关系相关的覆盖图的画法实际上是这一顺序关系的图表示, 且只要 x 覆盖 y 则 y 低于 x. 另外, 任意图表示都是一个顺序关系的覆盖图. 考虑图 4.11 中的图表示 b. 这一图表示定义由图 4.9b定义的关系(X, R)的覆盖图. 这里, $K=\{(1, 2), (1, 3), (2, 4), (3, 4)\}$. [251]

再一次考虑集合 X 上的顺序关系 R. 我们将使用 xSy 表示 xRy 且 $x\neq y$. 如果 xSy 或 ySx, 那么我们说 x 和 y 是**可比较的**(comparable), 记作 xCy. 另外, 如果既没有 xRy 也没有 yRx, 则我们说 x 和 y 是**不可比较的**(incomparable), 记作 xIy.

S, C 和 I 的图表示的含义也是一目了然的. xSy 当且仅当存在一条从 x 到 y 的下降链. 从 x 到 y 或 y 到 x 的下降链表明 xCy, 而对于 $x\neq y$, xIy 表明 x 和 y 之间不存在链, 或者仅有既不是严格上升也不是严格下降的链. 在图 4.10b 的图表示中, 我们马上看到 $3C7$, $3S7$, $\sim 2C6$ 和 $2I3$.

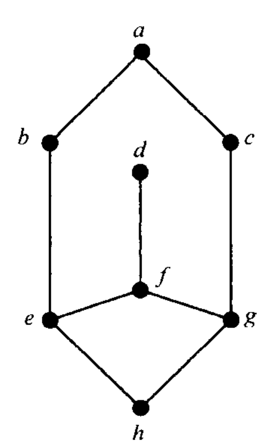


图 4.12　另一个图表示

一个元素 x 在有序集合(X, R)是**极大的**(maximal), 如果不存在 $y\in X$ 使得 ySx. 如果只存在一个极大元素, 那么这一元素称为**最大**(maximum)元素且记为$\hat{1}$. 我们设 $\max(X, R)$是(X, R)中极大元素的集合. 对于**极小**(minimal)元素、**最小**(minimum)元素以及 $\min(X, R)$也可作类似的定义. 如果最小元素存在, 则记为$\hat{0}$. 考虑图 4.12 中的图表示. 这里 d 是极大元素, h 是最小元素, $\max(X, R)=\{a, d\}$, 且 $\min(X, R)=\{h\}$(因为 h 是最小元素). 尽管每一个(在一个有限的集合上的)顺序关系至少有一个极小元素及一个极大元素[参见练习 24(a)], 但不一定有最大元素或最小元素.

4.2.3　线性序

在很多应用中, 我们寻找一个为各选项"定级"的顺序关系, 即它给出第一选择、第二选择、第三选择等等. 这样的顺序就是我们称为严格线性序的一个例子. 每当我们需要"排列"一组元素时, 我们就要为那个集合创建一个严格线性序. 患者在门诊等待看医生时, 学生需要排成一队时, 为某一频道的电视广播调度电视节目时, 都是严格线性序的例子. 当我们对可选的政治候选人定级别, 或根据"质量"为软件包定级别时, 我们的目标是生成一个严格线性序. 一个二元关系称为**严格线性序**(strict linear order), 如果它是传递、不对称、完全的. 图 4.13 所画的顺序关系就是严格线性序的例子(也使用**严格全序**(strict total order)和**严格简单序**(strict simple order)等说法). 一个严格线性序可能满足且将满足更多的性质, 但是定义它只需要这三个给定的性质(留给读者证明任意一个定义性质都不是多余的; 见练习 21). 事实上, 一个严格线性序还是反自反、反对称和负传递的. 不对称性蕴涵反自反性和反对称性(参见定理 4.3 的证明), 而传递性与完全性蕴涵负传递性(后者的证明留给读者; 参见练习 22). [252]



图 4.13　图 4.2 的有向图的一个生成子图

严格线性序的原型例子是任意实数集合上的关系$>$. 根据定理 4.1 和完全性, 对于有限集合 X 上的严格线性序 R, R 的图表示是由分布在一条垂线上的 X 的元素组成的, 即 aRb 当且仅当 a 在 b 的上方. 另一方面, 如果我们假设反自反性(没有环), 那么一个成垂线形状的图表示将总是一个严格线性序; 完全性是唯一需要核实的性质, 因为传递性和不对称性得自于反自反性假设

下的顺序关系的定义. 为了证明完全性，考虑 X 中的任意两个元素 $a \neq b$. 因为在垂线图表示上一个元素一定在另一个元素的上方，根据图表示的定义，较高元素与较低元素有关系"R". 因此，必须或者 aRb 或者 bRa. 因为画严格线性序(垂线图)无法说明，所以我们使用 Trotter [1992] 的表记法来简洁地描述严格线性序. 令 $L_S = [a_{i_1}, a_{i_2}, \cdots, a_{i_n}]$ 表示集合 $X=\{a_1, a_2, \cdots, a_n\}$上的严格线性序，其中只要在 L_S 中 a_{i_j} 超前 a_{i_k}，则有 $a_{i_j} S a_{i_k}$. 在这一表记法中，图 4.13 中的严格线性序由[纽约，迈阿密，亚特兰大，旧金山，洛杉矶]给出.

回想严格顺序中"严格"一词指的是这一关系是反自反的，而"非严格"的解释则是自反的. 我们通过反对称性、传递性和强完全性定义线性序(linear order). 容易证明，除了线性序是自反的之外，线性序和严格线性序有同样的性质. 特别地，线性序的原型例子是任意实数集合上的关系$\geqslant$. 严格线性序的记法$[a_{i_1}, a_{i_2}, \cdots, a_{i_n}]$也用于线性序 R，所做的唯一改变是对于所有的 j 有 $a_j R a_j$.

253

【例 4.7 字典序(Stanat 和 McAllister[1977])】 设 Σ 是一个有限的字母表，且设 R 是 Σ 上的严格线性序. 设 X 是 Σ 的串的集合，如下定义 X 上的字典序 S. 首先，如果 x 是 y 的前缀则有 xSy. 其次，假设 $x=zu$，$y=zv$ 且 z 是 x 和 y 的最长公共前缀. 这时，如果在严格线性序 R 中，u 的第一个符号超前 v 的第一个符号，则有 xSy. 例如，如果 Σ 是字母表$\{a, b, c, \cdots, z\}$，且 R 是通常的字母表顺序，那么 $abSabht$. 还有，$abdtSabeab$，因为我们有 $z=ab$，$u=dt$，$v=eab$，且 u 的第一符号 d 超前于 v 的第一个符号 e. 这一二元关系(X, S)对应于字典中所使用的一般顺序. 这一顺序关系定义一个严格线性序，因为它是传递的、不对称的且完全的(参见练习 15).

让我们继续这一情况，其中 Σ 是字母表$\{a, b, c, \cdots, z\}$，且 R 是通常的字母表顺序. 用 4.2.2 节的术语，如果 x 是 X 的任意元素，x 覆盖 xa，因为 $xSxa$ 且在 X 中不存在串 z 使得 xSz 且 $zSxa$. 另一方面，没有元素覆盖 xb. (为什么? 参见练习 17.)

这一例子有一个无限集合 X. 如果我们至多取给定长度的之内的所有串，那么就得到一个有限集合的例子. ■

4.2.4　弱序

当我们要对选择项定级为第一选择、第二选择等等时，我们可能希望有平级的选择. 接下来，我们考虑称为弱序的关系，它们很像线性关系，只不过允许有平级. 一个弱序(weak order)是一个传递且强完全的二元关系. 注意，因为没有假设反对称性，所以弱序可能不是顺序关系. 因为反对称性是线性序的一个定义性质，而弱序却没有必要假设有这一性质，所以，对于弱序 R，可能有 aRb 且 bRa 对 $a \neq b$. 在这种情况下，我们可以认为 a 和 b 在 R 中是平级的. 图 4.14 给出一个典型弱序的"图表示". 在我们已定义的图表示的意义下，这不是一个图表示. 这里，每一个元素有一个水平层次，同一水平层次上的所有元素 a 和 b 满足 aRb 且 bRa，除此之外，aRb 当且仅当 a 的层次比 b 高. 因此，在对应于图 4.14 的弱序中，R 由下式给出：

$$x_i R y_i \text{ 当且仅当 } x = y \text{ 或在字母表中 } x \text{ 超前 } y,$$

其中 x 和 y 是 a，b，c，d，e，或 f：例如，$b_2 R e_3$ 和 $a_1 R f_2$.

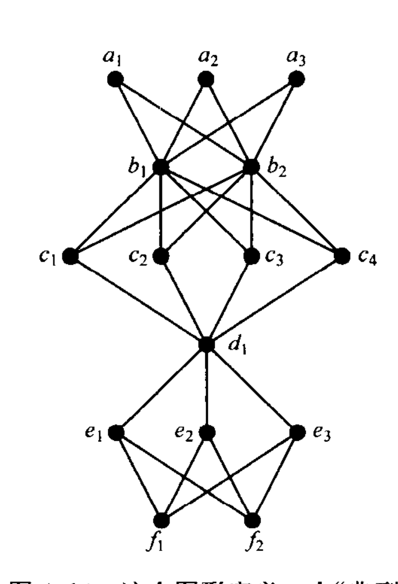


图 4.14　这个图形定义一个"典型"的弱序/严格弱序

你可以证明(参见 Roberts[1976，1979b])每一个(有限集合上的)弱序都是这样的.

有时候有必要考虑*严格弱序*(strict weak order)，严格弱序是与弱序相同的二元关系，只是对于同一水平层次上的元素 a 和 b，我们没有 aRb. 因此，在像图 4.14 的图中，我们有 x_iRy_j 当且仅当在字母表中 x 超前 y. 很容易看出，从这样一个“图表示”方式定义的关系 R 是不对称且负传递的. 相反，如果 X 是有限的，不对称且负传递的关系(X, R)可以被看成是以这样方式来自于像图 4.14 的一个“图表示”(参见 Roberts[1976，1979b]). 不对称且负传递的关系称为*严格弱序*(strict weak order). 不难证明，弱序与严格弱序之间的关联就如同线性关系与严格线性关系之间的关联一样. 它们之间的唯一差别就是处于同一水平层次的元素，或“平级”元素.

254

注意，尽管根据我们的定义弱序不一定是顺序关系，但是严格弱序却总是顺序关系(为什么). 还要注意，严格弱序只允许不可比较元素以特殊方式存在. 图 4.2 中的有向图定义一个严格弱序. 不难看出可以重画这一有向图：顶层是波士顿和纽约，接着是迈阿密，然后是亚特兰大和菲尼克斯，然后是旧金山，最后是洛杉矶. 从(4.5)式的函数 f 可以容易地得到这些层次. 回想一下，函数 f 是由(4.4)式定义的一个效用函数. 反之，如果一个严格优先选择关系 P 是由(4.4)式的效用函数 f 定义的，那么不难看到它是不对称且负传递的，即是一个严格弱序. 这一证明留作练习(练习 20).

因为同样的原因，图 4.2 的有向图定义一个严格弱序，图 4.5 的有向图也定义一个严格弱序. 如果像例 4.4 一样，令 $f(x)=r(x, q)$，那么我们有

$$xRy \Leftrightarrow r(x,q) > r(y,q) \Leftrightarrow f(x) > f(y),$$

这给出(4.4)式的一个等价形式.

255

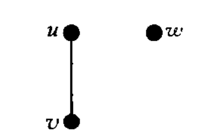


图 4.15 在严格弱序中看不到的图表示

不难证明图 4.3 和图 4.6 中的有向图不是严格弱序. 因此，不存在满足(4.4)式的函数. 为了弄明白其中的原因，我们考虑图 4.15 中的图表示. 注意，元素 u 和 w 是不可比较的，元素 w 和 v 也是不可比较的. 特别地，有 $\sim uRw$ 和 $\sim wRv$. 然而，有 uRv. 因此，这一图表示不是负传递的，也不可能是一个更大的负传递图表示的一部分. 这一图表示本质上是关于图表示不是负传递的定义. 一个顺序关系是严格弱序当且仅当它的图表示不包含图 4.15的图表示.

4.2.5 稳定婚姻[⊖]

假设 n 位男士和 n 位女士考虑婚姻问题. 在我们决定如何配成若干对夫妇之前，每一位男士和每一位女士都提供一个异性优先选择列表，一个严格线性序. 一个*稳定婚姻的集合*(set of stable marriage，或一个*稳定匹配*(stable matching))是男士和女士的配对(或匹配)，使得离开他们各自被指定的伴侣，并相互结婚，没有一位男士和一位女士境况会更好(就其选择来说). Gale 和 Shapley[1962]对这一问题及其若干变形作了介绍.

考虑 $n=4$ 且优先选择是由表 4.4 中所给出的严格线性序的情况. 注意，$M_1=\{m_1-w_4, m_2-w_3, m_3-w_1, m_4-w_2\}$是一个稳定婚姻的集合. 为了弄清楚其原因，注意 w_4 和 w_3 都跟她们的第一选择结婚；所以她们二人都不愿意与她们的配偶 m_1，m_2 离婚. 同样，m_3 和 m_4 也在其他女士中都得到他们的首选，即 w_1 和 w_2. 对于这一问题，共存在 10 种稳定匹配. 另外两种明显的稳定匹配是 $M_9=\{m_1-w_1, m_2-w_2, m_3-w_3, m_4-w_4\}$ 和 $M_0=\{m_1-m_4, m_2-w_3,$

⊖ 本小节的内容基于 Gusfield 和 Irving[1989]的工作.

表 4.4　大小为 4 的稳定婚姻问题的优先选择顺序（严格线性序）

男士的优先选择	女士的优先选择
m_1: $[w_1, w_2, w_3, w_4]$	w_1: $[m_4, m_3, m_2, m_1]$
m_2: $[w_2, w_1, w_4, w_3]$	w_2: $[m_3, m_4, m_1, m_2]$
m_3: $[w_3, w_4, w_1, w_2]$	w_3: $[m_2, m_1, m_4, m_3]$
m_4: $[w_4, w_3, w_2, w_1]$	w_4: $[m_1, m_2, m_3, m_4]$

256

m_3-w_2，$m_4-w_1\}$. 这两个都是稳定的，因为每一位男士（女士）都已得到他的（她的）第一选择. 10 种稳定匹配的完整列表是

$$M_0=\{m_1-w_4, m_2-w_3, m_3-w_2, m_4-w_1\}$$
$$M_1=\{m_1-w_4, m_2-w_3, m_3-w_1, m_4-w_2\}$$
$$M_2=\{m_1-w_3, m_2-w_4, m_3-w_2, m_4-w_1\}$$
$$M_3=\{m_1-w_3, m_2-w_4, m_3-w_1, m_4-w_2\}$$
$$M_4=\{m_1-w_2, m_2-w_4, m_3-w_1, m_4-w_3\}$$
$$M_5=\{m_1-w_3, m_2-w_1, m_3-w_4, m_4-w_2\}$$
$$M_6=\{m_1-w_2, m_2-w_1, m_3-w_4, m_4-w_3\}$$
$$M_7=\{m_1-w_2, m_2-w_1, m_3-w_3, m_4-w_4\}$$
$$M_8=\{m_1-w_1, m_2-w_2, m_3-w_4, m_4-w_3\}$$
$$M_9=\{m_1-w_1, m_2-w_2, m_3-w_3, m_4-w_4\}$$

给定所有 10 种稳定匹配的集合，某个人 x 可能较之稳定匹配 M_j 更偏爱 M_i，如果 x 偏爱他在 M_i 中的配偶而不是他在 M_j 中的配偶. 于是，我们可以如下定义面向男士的支配关系（man-oriented dominance relation）：M_i 支配 M_j，如果每一位男士或者较之 M_j 更偏爱 M_i，或者没有偏爱关系. 不难证明面向男士的支配是一个偏序. 它的图表示如图 4.16 所示（关于稳定婚姻问题的更全面讨论参见 12.8 节）.

图 4.16　稳定匹配上的面向男士的支配关系的偏序

本节练习

1. 下列关系中哪些是顺序关系？

257

(a)集合{1, 2, 3, 4}的子集的集合上的⊊.

(b)(X, P)，其中 $X=Re\times Re$ 且

$$(a,b)P(s,t) \text{ 当且仅当}(a>s \text{ 且 } b>t).$$

(c)(X, Q)，其中 X 是 n 维选项的集合，$f_1, f_2, \cdots, f_n$ 是 X 上的实值标量，Q 是由下式定义的：

$$aQb \Leftrightarrow [\text{对每个 } i, f_i(a)>f_i(b)].$$

(d)(X, Q)，其中 X 同(c)，且有

$$aQb \Leftrightarrow [\text{对每个 } i, f_i(a)\geqslant f_i(b) \text{ 且对某个 } i, f_i(a)>f_i(b)].$$

2. 练习 1 中的二元关系中哪些是严格偏序？
3. 练习 1 中的二元关系中哪些是线性序？
4. 练习 1 中的二元关系中哪些是严格线性序？
5. 练习 1 中的二元关系中哪些是弱序？
6. 练习 1 中的二元关系中哪些是严格弱序？
7. 下列各关系中哪些是线性序？

(a)集合 $X=\{a, b, c, d\}$ 上的关系{(a, a), (b, b), (c, c), (d, d), (c, d), (c, b), (c, a), $(d,$

$b)$, (d, a), $(b, a)\}$

(b)集合 $X=\{a, b, c, d\}$上的关系$\{(a, a), (a, b), (a, d), (b, b), (b, d), (c, a), (c, b), (c, d), (d, d)\}$

(c)集合 $X=\{a, b, c, d\}$上的关系$\{(a, a), (a, b), (a, c), (a, d), (b, b), (b, c), (b, d), (c, a), (c, c), (c, d), (d, d)\}$

(d)集合 $X=\{\lambda, \xi, \zeta, \varphi, \rho\}$上的关系$\{(\lambda, \lambda), (\lambda, \xi), (\lambda, \zeta), (\lambda, \varphi), (\xi, \xi), (\xi, \zeta), (\xi, \varphi), (\rho, \lambda), (\rho, \xi), (\rho, \rho), (\rho, \zeta), (\rho, \varphi), (\zeta, \zeta), (\zeta, \varphi), (\varphi, \varphi)\}$

8. 下列关系中哪些是严格线性序？

(a)集合 $X=\{a, b, c, d\}$上的关系$\{(c, d), (c, b), (c, a), (d, b), (d, a), (b, a)\}$

(b)集合 $X=\{a, b, c, d\}$上的关系$\{(a, b), (c, b), (a, d), (c, d), (b, d), (c, a), (b, c)\}$

(c)集合 $X=\{\lambda, \xi, \zeta, \varphi, \rho\}$上的关系$\{(\lambda, \delta), (\lambda, \xi), (\lambda, \zeta), (\lambda, \varphi), (\xi, \delta), (\xi, \zeta), (\xi, \varphi), (\xi, \xi), (\rho, \lambda), (\rho, \xi), (\rho, \delta), (\rho, \zeta), (\rho, \varphi), (\zeta, \delta), (\zeta, \varphi), (\varphi, \delta)\}$

(d)集合 $X=\{1, 2, 3, 4, 5\}$上的关系$\{(5, 1), (5, 2), (5, 4), (1, 2), (1, 4), (3, 5), (3, 1), (3, 2), (3, 4), (2, 4)\}$

9. 下列关系中哪些是弱序？

(a)集合 $X=\{a, b, c, d\}$上的关系$\{(a, a), (b, b), (c, c), (d, d), (b, a), (b, c), (b, d), (c, a), (c, d), (a, d)\}$

(b)集合 $X=\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$上的关系$\{(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (1, 9), (2, 2), (2, 5), (2, 6), (2, 7), (2, 8), (2, 9), (3, 3), (3, 5), (3, 6), (3, 7), (3, 8), (3, 9), (4, 4), (4, 5), (4, 6), (4, 7), (4, 8), (4, 9), (5, 5), (5, 7), (5, 8), (5, 9), (6, 6), (6, 7), (6, 8), (6, 9), (7, 7), (7, 9), (8, 8), (8, 9), (9, 9)\}$

(c)集合 $X=\{\Gamma, \Delta, \Psi, \Upsilon, \Omega\}$上的关系$\{(\Gamma, \Delta), (\Delta, \Psi), (\Gamma, \Upsilon), (\Delta, \Upsilon), (\Gamma, \Omega), (\Upsilon, \Psi), (\Gamma, \Psi), (\Psi, \Omega), (\Delta, \Omega), (\Upsilon, \Omega)\}$

10. 练习 9 的二元关系中哪些是严格弱序？ 258

11. 画出对应于下列各图中的严格偏序的图表示：

(a)图 4.3　　(b)图 4.4　　(c)图 4.6

12. 使用与图 4.14 同样的方法，画出图 4.5 中的严格弱序的图表示，使得“平级”元素位于相同水平层次.

13. 假设 $X=\{1, 2, 3, 4\}$且

$$R=\{(1,1),(2,2),(3,3),(4,4),(1,3),(1,4),(2,3),(2,4),(3,4)\}.$$

(a)证明(X, R)是一个偏序.

(b)画出(X, R)的图表示.

(c)寻找与(X, R)相关的覆盖关系(X, K).

14. 假设 $X=\{t, u, v, w, x, y, z\}$且

$$R=\{(t,u),(t,v),(u,v),(w,v),(w,x),(w,y),(w,z),(x,v),(x,z)(y,v),(y,z),(z,v)\}.$$

(a)证明(X, R)是严格偏序.

(b)画出(X, R)的图表示.

(c)寻找与(X, R)相关的覆盖关系(X, K).

15. 证明例 4.7 中的二元关系(X, S)是：

(a)传递的　　(b)不对称的　　(c)完全的

16. 根据例 4.7，对于严格线性序(X, S)写出 S，其中 X 是长度至多为 4 的所有串的集合，且

(a)$\Sigma=\{a, b\}$如果 aRb.　　(b)$\Sigma=\{a, b, c\}$如果 aRb, aRc, bRc.

17. 根据例 4.7，如果 $\Sigma=\{a, b, c, \cdots, z\}$且 R 是普通字母表顺序，解释为什么 $xb\in X$ 没有覆盖.

18. 假设 $X=\{a, b, c, d, e, f\}$且

$$R=\{(a,c),(a,f),(b,c),(b,f),(d,a),(d,b),(d,c),(d,e),(d,f),(e,c),(e,f)\}.$$

(a)证明：(X, R)是严格弱序.

(b)画出(X, R)的图表示.

19. 考虑严格线性序 $L_S=[x_1, x_2, \cdots, x_n]$.

(a)寻找 L_{S-1}. (b)寻找 $L_S \cap L_{S-1}$. (c)寻找 $L_S \cup L_{S-1}$.

20. 如果(X, P)是由(4.4)式的效用函数 f 定义的严格优先选择关系，证明 P 是不对称且负传递的.

21. 证明下面三个性质中的任意两个都不蕴涵第三个性质：传递性、完全性、不对称性.

22. 证明：传递且完全的二元关系是负传递的.

259 23. 重画图 4.9 中的有向图 c 以证明这是一个顺序关系. 即重新画出这一有向图使得所有弧都是下降的.

24. (a)证明；每一个(有限集合上的)顺序关系至少有一个极大元素和一个极小元素.

(b)证明：每一个(严格)线性序有最大元素和最小元素.

25. (a)严格偏序的逆 R^{-1} 一定是严格偏序吗(对于逆的定义，参见 4.1 节练习 7)?

(b)偏序的逆 R^{-1} 一定是偏序吗?

26. 证明：每一个严格弱序是严格偏序.

27. 如果(X, R)是严格弱序，通过下式定义 X 上的关系 S：

$$aSb \text{ 当且仅当}(aRb \text{ 或 } a=b).$$

证明：(X, S)是偏序.

28. 画出图 4.14 定义的严格偏序的逆的图表示(对于逆的定义，参见 4.1 节练习 7).

29. (a)严格弱序的逆一定是严格弱序吗? 为什么?

(b)严格偏序的逆一定是严格偏序吗? 为什么?

30. (a)严格弱序的补一定是严格弱序吗(关于补的定义，参见 4.1 节练习 8)?

(b)严格偏序的补一定是严格偏序吗?

31. 证明：4.2.5 节的面向男士支配关系是一个偏序.

32. 考虑 4.2.5 的稳定匹配集合.

(a)写出稳定匹配集合上的面向女士支配关系的定义.

(b)画出这一稳定匹配集合上的面向女士支配关系的图表示.

(c)比较面向男士支配的图表示和面向女士支配的图表示?

33. (a)解释为什么

$$\{m_1-w_1, m_2-w_4, m_3-w_2, m_4-w_3\}$$

不是表 4.4 的优先选择顺序的稳定婚姻集合.

(b)对于表 4.4 的优先选择顺序，有多少婚姻集合不是稳定的?

(c)证明：$M_0, M_1, \cdots, M_9$ 是表 4.4 的优先选择顺序的全部稳定婚姻集合.

4.3 偏序的线性扩展

4.3.1 线性扩展和维数

【例 4.8 推理小说与线性扩展】 在写一部优秀的谋杀推理小说时，作者给出若干线索帮助读 260 者弄明白"谁干了这件事". 某些线索的理解依赖于前面的线索. 考虑图 4.17. 这一有向图的顶点代表某本小说的线索和"解"，弧代表一条线索与另一条线索的依赖关系. 例如，从线索 3 到线索 9 的弧表示要明白线索 9 需要线索 3. 显然，我们可以假设传递性，所以我们不必画出线索间的所有弧. 反对称性也一定成立，所以我们的有向图定义一个顺序关系. 从某些线索到解顶点的弧表示明白这一推理小说所需要的那些线索. 因此，根据传递性，弄明白这一推理小说需要所有的线索.

作者所面临的任务是以什么样的顺序在小说中连贯地描述线索．说到连贯，我们指的是所有线索都出现在所有它依赖的线索之后．依赖于这一有向图，存在很多描述这些线索的方式．图4.18给出四种描述图4.17的有向图中的线索的方式．同样，在这些有向图中都假设了传递性．图4.18中的有向图说明了线性扩展的思想．

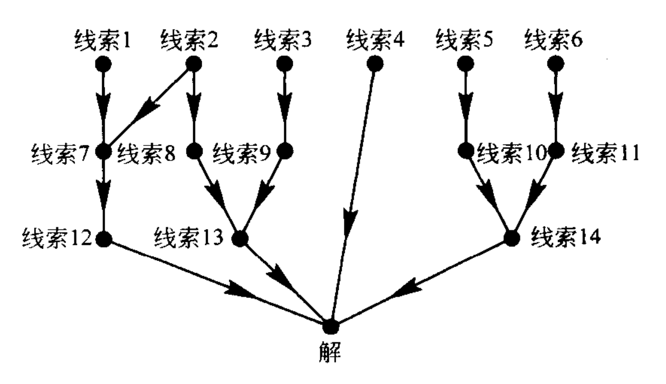


图4.17　一部小说的"线索"有向图

考虑同一集合 X 上的严格偏序 R 和 S．如果只要 aSb 就有 aRb，那么 R 是 S 的扩展(extension)．换一种说法，R 是 S 的扩展，如果所有定义 S 的序对都在 R 中．如果 R 是严格线性序，那么 R 称为 S 的线性扩展(linear extension)．事实上，图4.17的有向图是严格偏序(我们省略了传递性蕴涵的弧)，而且图4.18中的每一个有向图都给出这一顺序关系的一个线性扩展．线性扩展恰好就是推理小说作者要考虑的事情．由于书中文字的线性特征，必须按一个线性扩展所给的顺序展开这些线索．■

【例4.9 拓扑排序】 当我们把一个严格偏序 (X, R) 输入到一台计算机时，自然要产生严格偏序的线性扩展．我们把 X 的元素按序列顺序输入，并可能希望以保持这一严格偏序的方式输入它们．这一过程扩展为寻找使用整数 $1, 2, \cdots, n$ 标记任意有向图的顶点，使得 D 的每一个条弧都是从较小标签的顶点到较大标签的顶点．这一过程称为*拓扑排序*(topological sorting)．我们将在11.6.2节讨论这一问题，在那里我们将描述这一思想在项目计划、调度和设施设计等方面的应用．■

261

下面的定理说明线性扩展不是偶然发生的．

定理4.4(Szpilrajn的扩展定理[1930]) *每一个严格偏序都有线性扩展．另外，如果 (X, R) 是一个严格偏序且对于不同的 $x, y \in X$ 有 xIy(即 x 与 y 是不可比较的)，那么至少存在一个具有 xLy 的线性扩展 L．*

不难证明下面的算法可以寻找给定严格偏序的线性扩展．实际上，这一算法可以用于寻找严格偏序的所有线性扩展．该算法使用子顺序的概念．如果 (X, R) 是一个严格偏序，D 是它相应的有向图，且 $Y \subseteq X$，那么由 Y 生成的子图也是一个严格偏序的有向图，因为它是不对称且传递的．我们称这一关系为一个*子顺序*(suborder)，并记为 (Y, R)(从技术上讲，这不是一个适宜的记法．我们指的是包含在 R 中的 $Y \times Y$ 的所有有序对)．图4.13是由下

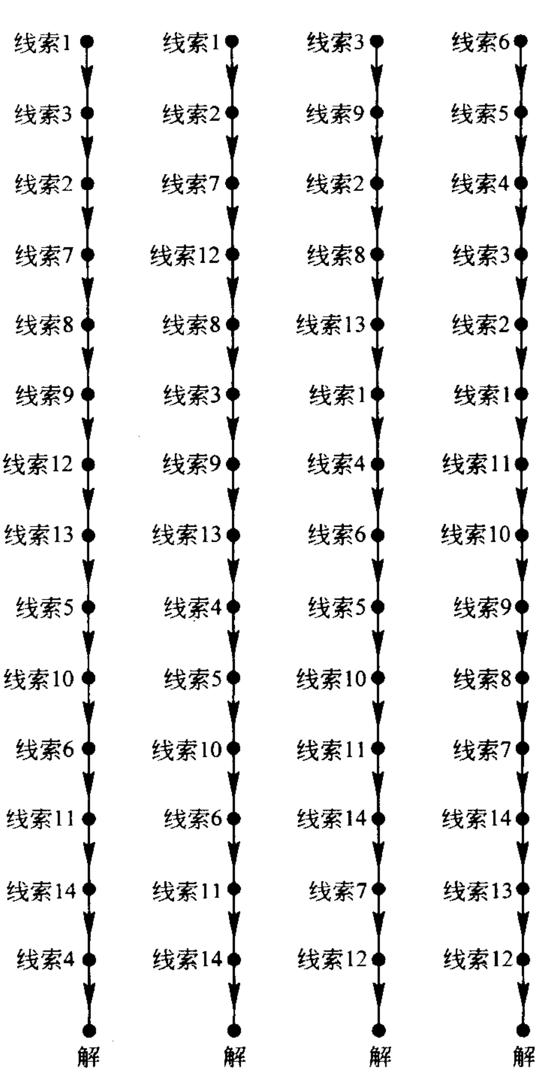


图4.18　可能的线索表示模式

面集合生成的图 4.2 中的顺序关系的子顺序：

$$Y = \{纽约,迈阿密,亚特兰大,旧金山,洛杉矶\}.$$

算法 4.1：线性扩展

输入：严格偏序 (X, R).

输出：(X, R) 的一个线性扩展.

步骤 1. 令 $m = |X|$.

步骤 2. 寻找 (X, R) 中的一个极小元素 x.

步骤 3. 令 $X = X - x$, $R = R \cap \{(X-x) \times (X-x)\}$, 且 $b_m = x$.

步骤 4. 把 m 递减 1. 如果现在 m 是 0，停步并输出有向图 B，其顶点集合是 X，且当 $i < j$ 时 (b_i, b_j) 是弧. 如果不是，返回步骤 2.

关于这一算法有几点说明：(1)根据定理 4.2，严格偏序总有极小元素，步骤 2 中需要这一元素. (2)$X-x$ 和由步骤 3 重新定义的 R 是我们记为 $(X-x, R)$ 的子顺序. 它也是一个严格偏序，因此有极小元素. (3)容易看出，B 是 (X, R) 的线性扩展.

为了说明算法 4.1，考虑图 4.19 中的严格偏序. 在步骤 1，令 $m=4$，在步骤 2，我们可以选择极小元素 a_4. 然后，在步骤 3，我们考虑除去 a_4 的严格偏序，并令 $b_4 = a_4$. 接下来，我们选择极小元素 a_3，令 $b_3 = a_3$，且考虑除去 a_3(及 a_4)的严格偏序. 然后我们可以选择极小元素 a_2，令 $b_2 = a_2$，且考虑除去 a_2(以及 a_3 和 a_4)的严格偏序. 最后，我们选择剩余的(极小)元素 a_1，并令 $b_1 = a_1$. 步骤 4 定义的有向图 B 对应于图 4.19 中的线性扩展 L_1.

262～263

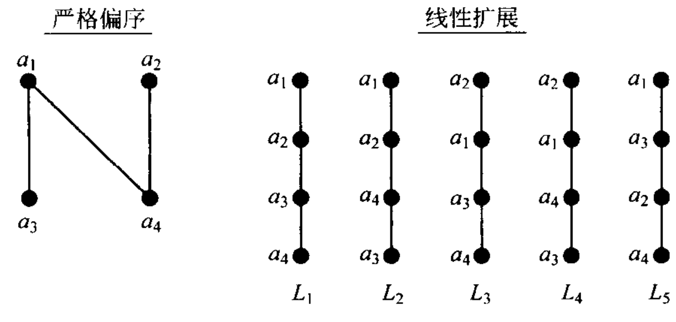


图 4.19　四个元素上的一个严格偏序以及它的线性扩展

让我们考虑严格偏序 (X, R) 的所有线性扩展的集合 $\mathcal{F}$. 如果 xRy，则在 $\mathcal{F}$ 的每一个线性扩展中这一结论都成立. 如果 xIy，那么，根据 Szpilrajn 的扩展定理，在 $\mathcal{F}$ 中至少有一个 x 在 y 之上的线性扩展和一个 y 在 x 之上的线性扩展. 因此，我们有下面的结果.

定理 4.5　如果 (X, R) 是一个严格线性序，且 $\mathcal{F}$ 是 (X, R) 的线性扩展的集合，那么

$$(X,R) = \bigcap_{L \in \mathcal{F}} L.$$

因为严格偏序是它的所有线性扩展的交集，所以一个很自然的问题是：需要多少个线性扩展才能使其交集是原来的严格偏序？给定严格偏序 (X, R)，其交集是 (X, R) 的最小线性扩展数目称为 (X, R) 的维数(dimension)，写作 $\dim(X, R)$(这一概念是 Dushnik 和 Miller[1941]提出的). 图 4.19 给出严格偏序 (X, R) 的所有五个线性扩展，L_1，L_2，L_3，L_4，L_5(回想一下，对于四个元素的集合，存在 4! 个元素的顺序. 在这种情况下，4! =24 个可能的顺序中只有 5 个是线性扩展). 因为 (X, R) 不是线性序，所以 $\dim(X, R) > 1$，而且根据定理 4.5，有 $\dim(X, R) \leqslant 5$. 容易看到，$L_4 \cap L_5$ 等于 (X, R)[例如，在 (X, R) 中有 $a_3 I a_4$，在 L_5 中有 $a_3 S a_4$，而在 L_4 中有 $a_4 S a_3$]. 因此，$\dim(X, R) = 2$. 寻找任意顺序关系的维数不是一个简单的问题. 给定顺序关系 (X, R)，Yannakakis[1982]证明对于每一个固定的 $t \geqslant 3$，检验 $\dim(X, R) \leqslant t$ 是 NP 完全的. 关于维数的概述文章参见 Trotter[1996]. 根据 Yannakakis 的结果，与维数相关的大多数结果都对有序集合的特殊类达到边界值或准确值. 我们在下节给出其中的几个.

264

【例 4.10 多维效用函数】 假设(X, P)是例 4.1 的严格优先选择关系. 在例 4.6 中，我们讨论了一个函数 f，它对 X 中的每一个 a 度量值 $f(a)$，使得当且仅当 a 的值大于 b 的值时，与 b 相比严格优先选择 a，即

$$aPb \leftrightarrow f(a) > f(b).$$

如果我们使用若干特征或维数，比如说值 $f_1(a)$，美丽 $f_2(a)$，质量 $f_3(a)$等，情况将如何呢？我们可以表达关于 a 胜过 b 的严格优先选择，当且仅当相对于每一个特征 a 都比 b 更好；即有

$$aPb \leftrightarrow [f_1(a) > f_1(b)] \& [f_2(a) > f_2(b)] \& \cdots \& [f_t(a) > f_t(b)]. \tag{4.6}$$

这样的情况经常出现在软件和硬件的比较中. 例如，我们可以根据成本、速度、精确度以及易用度给两个不同的软件包做等级评估(评估结果为：0=不好，1=一般，2=好，3=非常好). 于是我们可以明确一个软件包严格优先于另一个软件包，当且仅当在每一个“维数”上，前一个软件包的分数都更高. (精确地说，只有当使用 1/成本而不是成本才行. 为什么)我们也可以把我们的决定局限在那些没有其他严格优先于它的软件包上. Fenton 和 Pfleeger[1997, p. 225]描述了一个详细的例子.

如果 P 是使用(4.6)式定义的，容易看到，(X, P)是严格偏序. 其逆问题在优先选择理论中是一个很重要的问题. 假设我们有一个严格偏序(X, P). 你能够寻找到函数 f_1, f_2, …, f_t，其中每一个 f_i 给 X 中的每一个 a 指定一个实数，使得(4.6)式成立吗？事实上，如果(X, P)有维数 t，且有 $P=\bigcap_{i=1}^{t} L_i$，那么我们可以定义 f_i，使得

$$aL_ib \leftrightarrow f_i(a) > f_i(b).$$

于是(4.6)式成立. 因此，对于足够大的 t，我们总能找到 f_1, f_2, …, f_t，而且我们能够寻找到这样的函数的最小的数字 t 至多是 P 的维数. 事实上，这样的最小数等于这一维数，除非当这一维数等于 2 时，在这种情况下，我们也许能够找到满足(4.6)式的函数 f_1. 例如，(X, P)是严格弱序时出现这一情况. 关于严格偏序与由(4.6)式给出的优先选择的多维模型之间的联系可以参见 Baker, Fishburn 和 Roberts[1972]. 关于更多的多维效用函数的讨论，参见 Keeney 和 Raiffa[1993]、Vincke[1992]. ■

4.3.2 链和反链

在严格偏序(X, R)中，同时是严格线性序的子顺序称为*链*(chain). 因此，链是(X, R)的子顺序(Y, R)，其中 Y 中的任意两个不同元素都可以比较. 链(Y, R)的*长度*(length)等于 $|Y|-1$，这一数值等于(Y, R)的图表示中边的数量. 另一方面，不存在可比元素的子顺序被称为*反链*(antichain). 在图 4.14 中的图表示所给的严格偏序中，$L_1=\{a_1, c_2, d_1\}$和 $L_2=\{a_3, b_1, c_2, d_1, e_1, f_2\}$是链的例子，而$(Y, R)$，其中$Y=\{c_1, c_3, c_4\}$，是一个反链的例子. 术语“链”和“反链”有时候只是指严格偏序的子集而不是子顺序. 一个链或反链是极大的，如果它不是更长的链或反链的一部分. 注意，链 L_1 不是极大的，但是 L_2 是极大的. Y 不是极大反链，但是 $Y\cup\{c_2\}$是极大的.

265

关于链与反链之间关系的最著名的两个定理是迪尔沃斯(Dilworth)定理.

定理 4.6(Dilworth[1950]) *如果(X, R)是严格偏序且其最长链中的元素个数是 j，那么存在 j 个反链 X_1, X_2, …, X_j 使得*

$$X = X_1 \cup X_2 \cup \cdots \cup X_j,$$

而且对于 $i\neq k$，有

$$X_i \cap X_k = \varnothing.$$

证明 令 $X_1=\max(X, R)$，$X_2=\max(X-X_1, R)$，$X_3=\max(X-X_1-X_2, R)$，…. 持

续这一过程直到 $X-X_1-X_2-\cdots-X_p=$空集为止. 根据极大元素的定义，每一个 X_i 都是反链，而且对于 $i\neq k$，有

$$X_i \cap X_k = \varnothing.$$

为了完成这一证明，我们还需要证明 $j=p$.

最长链中的每一个元素一定在不同的反链 X_i 中. 因此，$p\geqslant j$. 考虑 X_i 和 X_{i+1}. 对于 X_{i+1} 中的每一个元素 x，在 X_i 中存在一个元素 y，使得 yRx. 否则，x 必定是 $X-X_1-X_2-\cdots-X_{i-1}$ 中的极大元素，且在 X_i 中. 因此，我们使用每一个 X_i 的一个元素构成一个链，所以有 $p\leqslant j$. ■

下面定理的证明类似，留作练习(练习 19).

定理 4.7(Dilworth[1950]) 如果(X, R)是严格偏序且其最大反链的大小是 j. 那么存在 j 条链 $C_1, C_2, \cdots, C_j$ 使得

$$X = C_1 \cup C_2 \cup \cdots \cup C_j,$$

而且对于 $i\neq k$，有

$$C_i \cap C_k = \varnothing.$$

266

【例 4.11 子集包含】 令 Δ 是集合$\{1, 2, \cdots, n\}$. 考虑 Δ 的所有子集的集合 S 上的严格偏序 $\subsetneq$. 互不包含的 Δ 的子集的最大集合是什么？这相当于问$(S, \subsetneq)$中的最大反链是什么.

显然，对于任意给定的 k，大小为 k 的所有子集的集合是大小为$\binom{n}{k}$的反链. 其中最大的反链是什么呢？容易证明，当 k 是 n 的"一半"时出现这一类型的最大反链(参见 2.7 节练习 26). 即对于 $0\leqslant k\leqslant n$，有

$$\binom{n}{k}\leqslant\binom{n}{\left\lfloor \frac{n}{2}\right\rfloor}. \tag{4.7}$$

因此，存在一个大小为$\binom{n}{\left\lfloor \frac{n}{2}\right\rfloor}$的反链. 存在更大的反链吗？Sperner[1928]证明了不存在更大的反链. 考虑$(S, \subsetneq)$中的极大链. 一个极大链是由空集、大小为 1 的一个子集、大小为 2 的一个子集……最后是这一集合本身组成的. 对于大小为 1 的子集，有 n 种可能的选择. 大小为 2 的子集必须包含前面选出的大小为 1 的子集. 因此，对于这一子集中的第二元素，存在 $n-1$ 种可能的选择. 使用这一方法，继续构建极大链，我们看到存在 $n!$ 个极大链. 假设 A 是一个反链. 如果 $s\in A$ 且 $|s|=k$，那么 s 属于 $k!(n-k)!$ 个极大链中(原因参见练习 14). 因为对于任意的链，A 中至多有一个元素属于该链，所以我们得到

$$\sum_{s\in A} |s|!(n-|s|)! \leqslant n!.$$

用 $n!$ 除以不等式两边，得到

$$\sum_{s\in A}\frac{1}{\binom{n}{|s|}}\leqslant 1. \tag{4.8}$$

结合(4.7)式和(4.8)式得到

$$\sum_{s\in A}\frac{1}{\binom{n}{\left\lfloor \frac{n}{2}\right\rfloor}}\leqslant 1. \tag{4.9}$$

因为(4.9)式的左边的被加数中没有 s，所以

$$\sum_{s\in A}\frac{1}{\binom{n}{\lfloor\frac{n}{2}\rfloor}}=\frac{1}{\binom{n}{\lfloor\frac{n}{2}\rfloor}}\sum_{s\in A}1=\frac{1}{\binom{n}{\lfloor\frac{n}{2}\rfloor}}\mid A\mid.$$

把这一等式与(4.9)式结合起来得到

$$\frac{\mid A\mid}{\binom{n}{\lfloor\frac{n}{2}\rfloor}}\leqslant 1 \quad 或 \quad \mid A\mid\leqslant\binom{n}{\lfloor\frac{n}{2}\rfloor}.$$

267

因此，大小为$\lfloor\frac{n}{2}\rfloor$的所有子集形成严格偏序$(S,\subsetneq)$中的最大反链：大小为$\binom{n}{\lfloor\frac{n}{2}\rfloor}$的一个反链. ■

链和反链的定义产生两个参数，宽(width)和高(height)，这两个参数可以用于界定一个严格偏序的维数. (X, R)的宽$W(X, R)$是最大尺寸反链的大小，而其高度$H(X, R)$等于最大尺寸链的长度. 图4.20中的严格偏序给出这些参数的例子.

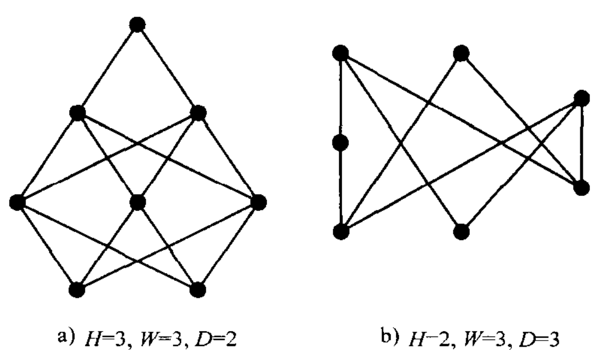


图4.20 严格偏序和它们的高H、宽W，以及维数D

我们将使用下面的引理及严格偏序的宽度界定其维数的上界.

引理4.1(Hiraguchi[1955]) *设(X, R)是严格偏序，且$C\subseteq X$是一个链. 那么，存在R的线性扩展L'_C，使得对于每个$x, y\in X$，若$x\in C$，$y\notin C$，且xIy，则有xL'_Cy.*

证明 使用算法4.1产生L'_C，尽可能总选择不在C中的元素. 在这些情况下生成的线性扩展一定满足这一引理的必要条件. ■

定理4.8(Dilworth[1950]) *给定严格偏序(X, R)，有*

$$\dim(X,R)\leqslant W(X,R).$$

证明 设$w=W(X, R)$. 由定理4.7，存在w个链$C_1, C_2, \cdots, C_w$，使得$X=C_1\cup C_2\cup\cdots\cup C_w$且对于$i\neq k$有$C_i\cap C_k=$空集. 我们将使用这些链构建$w$个线性扩展$L_1, L_2, \cdots, L_w$，它们的交集等于$(X, R)$.

268

令L_i等于根据引理4.1使用链C_i生成的线性扩展L'_{C_i}. 因为链$C_1, C_2, \cdots, C_w$不相互重叠，X的任意元素在且仅在一个链中. 如果xIy，那么一定存在包含x和y的不同链. 分别设这两个不同的链为C_j和C_k. 那么L_j有xL_jy而L_k有yL_kx. 这与线性扩展包含R的事实同时保证有$(X, R)=\bigcap_{i=1}^{w}L_i$. ■

考虑图4.21中的两个严格偏序S_5和C_5，每一个都是10个元素的集合上的严格偏序. S_5的宽等于5. 所以根据定理4.8，S_5的维数至多是5. 事实上，它等于5，这证明可以达到定理4.8的边界. 为了证明这一点，我们将证明没有四个线性扩展的交可以产生S_5. 假设$L_1\cap L_2\cap L_3\cap L_4=S_5$. 注意，对于$i=1, 2, 3, 4, 5$，有$x_iIy_i$. 因此，至少在一个线性扩展中，$y_i$超前$x_i$. 根据鸽巢原理，至少有两个这样的超前关系必定在同一个线性扩展中. 不失一般性，假设在L_1中y_1超前x_1，y_2超前x_2，因为在S_5中x_2超前y_1，x_1超前y_2，这在L_1中也一定为真. 因此，在L_1中，我们有$y_1L_1x_1$，$x_1L_1y_2$，$y_2L_1x_2$，$x_2L_1y_1$，传递性表明有$y_1L_1y_1$，

矛盾.

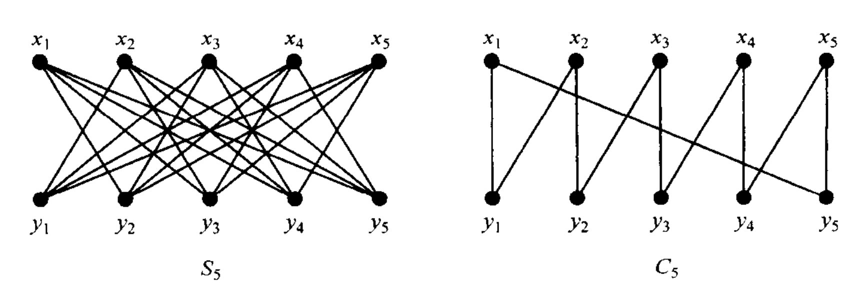


图 4.21 严格偏序 S_5 和 C_5

S_5 只是维数等于宽度的所有严格偏序类中的一个例子. 扩展 S_5，对于 $n\geqslant 3$，设 $S_n=(X, R)$ 是一个严格偏序，且 $X=\{x_1, x_2, \cdots, x_n, y_1, y_2, \cdots, y_n\}$. S_n 的极大元素是 $\{x_1, x_2, \cdots, x_n\}$，极小元素是 $\{y_1, y_2, \cdots, y_n\}$，这两个集合都是反链，且 x_iRy_i 当且仅当 $i\neq j$. S_n 将有维数 n 且宽度为 n. 其证明类似于 S_5 中的相关证明.

对于其他情况，定理 4.8 中的边界可能远离希望的值. 图 4.21 中的第二个严格偏序 C_5，有宽度 5，而维数为 3. 这些事实的证明留作练习[练习 10(a)和 10(b)]. Baker，Fishburn 和 Roberts[1972]使用术语冠(crown)表示这种类型的严格偏序. 一般地，他们定义冠顺序关系 $C_n=(X, R)$，$n\geqslant 3$. 设 $X=\{x_1, x_2, \cdots, x_n, y_1, y_2, \cdots, y_n\}$. C_n 的极大元素是 $\{x_1, x_2, \cdots, x_n\}$，其极小元素是 $\{y_1, y_2, \cdots, y_n\}$，这两个集合是反链，且 x_iRy_i 当且仅当 $i=j$ 或 $j\equiv i-1$ (模 n). C_n 有维数 n 和宽度 3. 这一实事的证明留作练习(练习 16). 在练习中将给出维数的其他边界.

269

4.3.3 区间顺序

很多重要的严格偏序的维数都已计算出来. 我们以对一类非常重要的严格偏序，区间顺序的维数的讨论来结束本节. 为了理解区间偏序，想象存在一个你所做选择的选项的集合. 你不知道每一个选项 a 的精确值，但是你可以通过一个闭区间 $J(a)=[\alpha(a), \beta(a)]$所给的可能值的范围来进行评估. 于是，你优先选择 a 而不选择 b，当且仅当你确定 a 的值比 b 的值大，即当且仅当 $\alpha(a)>\beta(b)$. 容易证明(练习 20)，相应的有向图给出一个严格偏序，即它是不对称且传递的(在这一有向图中，顶点是一族闭实区间，当且仅当 $a>d$ 时有一条从区间$[a, b]$到区间$[c, d]$的弧). 以这种形式出现的任意严格偏序称为区间顺序(interval order). 区间顺序的概念是 Fishburn[1970a]提出的.

Fishburn[1970a]证明了下面的定理.

定理 4.9 *有向图 $D=(V, A)$是区间顺序，当且仅当 D 没有环，而且只要$(a, b)\in A$ 且$(c, d)\in A$，那么或者$(a, d)\in A$ 或者$(c, b)\in A$.*

为了说明定理 4.9，注意图 4.21 的严格偏序 C_5 不是区间顺序，因为(x_1, y_1)和(x_3, y_3)是 C_5 的弧，但是(x_1, y_3)和(x_3, y_1)不是.

虽然区间顺序本质上有某种一维性，但是，在区间顺序的研究中，令人惊奇的是作为严格偏序其维数可以任意大(这是下面的定理的内容，其证明使用了 2.19.3 节的拉姆齐理论的扩展). 这意味着如果优先选择以定义区间顺序的很自然的方式出现，那么我们可能需要非常大的维数或特性来解释例 4.10 中的(4.6)式意义下的优先选择.

定理 4.10(Bogart，Rabinovitch 和 Trotter[1976]) *存在任意大维数的区间顺序.*

本节练习

1. 图 4.18 中的四个线性扩展的交集等于图 4.17 的顺序关系吗?

270

2. (a)寻找两个线性扩展，其交集是图 4.20a 的严格偏序.

   (b)寻找三个线性扩展，它们的交集是图 4.20b 的严格偏序.

3. 分别使用函数 f_1，f_2，f_3 和 f_4，依据"1/成本"、"速度"、"精确性"和"易用性"评估六个软件包(SP)A，B，C，D，E 和 F. 假设已收集到下面的数据：

SP	f_1	f_2	f_3	f_4
A	1	1	2	2
B	0	0	1	0
C	3	3	2	3
D	3	2	3	3
E	0	1	2	1
F	2	3	3	2

   (a)使用(4.6)式确定严格优先选择关系 P.

   (b)定义一个新关系 P'，其中 $aP'b$ 当且仅当 a 的分数在 4 个评估函数中有 3 项高于 b. 确定 P'.

4. 确定下列严格偏序的维数：

   (a)图 4.22　(b)图 4.3a　(c)图 4.23b

   (d)图 4.23c　(e)图 4.24 的 P_1　(f)图 4.24 的 P_2

5. (a)给出正好有两个线性扩展的严格偏序的例子.

   (b)是否有可能给出正好有三个线性扩展的严格偏序的例子?

   (c)如果严格偏序(X, R)正好有两个线性扩展，寻找 $\dim(X, R)$.

6. (a)对于图 4.21 中的顺序关系 S_5，使用算法 4.1 来寻找一个线性扩展. 说明这一算法的每一步的结果.

   (b)对于 S_5 存在多少个线性扩展?

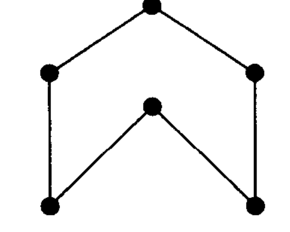


图 4.22　一个严格偏序

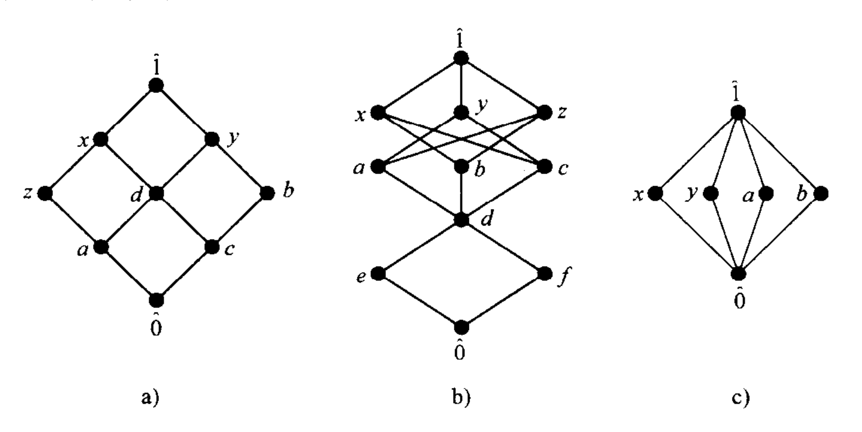


图 4.23　严格偏序(它们也是格)

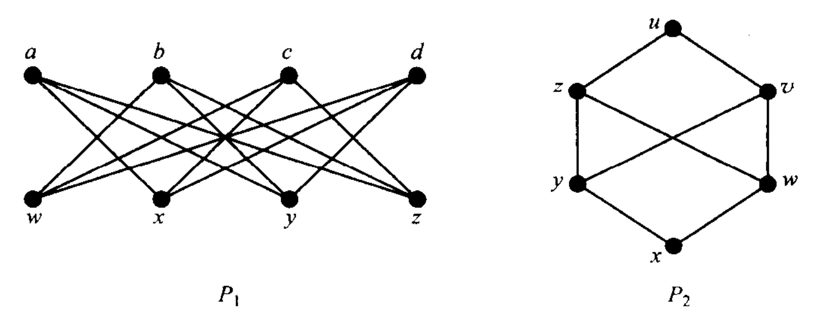


图 4.24　两个严格偏序

7. 确定严格偏序(X, R)的所有线性扩展的集合，其中$X=\{1, 2, 3, 4, 5, 6\}$，$R=\{(1, 3), (1, 4), (2, 3), (2, 4), (3, 4), (1, 5), (2, 5), (3, 5), (4, 5), (1, 6), (2, 6), (3, 6), (4, 6)\}$.
8. 确定图 4.23 中的每一个严格偏序的高和宽.
9. (a)设 C 是图 4.23a 的严格偏序中的链$[\hat{1}, d, a]$. 如引理 4.1，确定线性扩展 L'_C.

271 (b)对于 $C=[\hat{1}, x, d, a]$重复(a).

(c)对于图 4.23b 的严格偏序中的 $C=[y, d, \hat{0}]$重复(a).
(d)对于图 4.23c 的严格偏序中的 $C=[x]$重复(a).
10. 对于图 4.21 中的严格偏序 C_5：
(a)证明它的宽度是 5.
(b)证明它的维数是 3.
(c)证明 C_5 的任意子顺序有维数 2.
11. 如果 X 是集合$\{1, 2, 3\}$的所有子集的集合，且 R 是严格偏序$\subsetneq$，证明(X, R)有维数 3(Komm[1948]证明了一个集合 X 的子集的集合上的严格偏序$\subsetneq$有维数$|X|$).
12. Hiraguchi[1955]证明了如果(X, R)是一个严格偏序，且$|X|$是有限的并至少是 4，那么 $\dim(X, R)\leqslant|X|/2$. 证明维数可以小于$\lfloor|X|/2\rfloor$.
13. 证明每一个严格弱序的维数至多为 2.
14. 假设 A 是例 4.11 中的子集包含顺序中的一个反链. 如果 $s\in A$ 且$|s|=k$，证明 s 属于$k!(n-k)!$个极大链.
15. 回想 178 页上严格偏序 S_n 的定义. 证明 S_n 的维数和宽度都等于 n.
16. 回想 178 页上的冠严格偏序 C_n 的定义. 证明 C_n 的维数等于 3，宽度等于 n.
17. 对于图 4.24 中的严格偏序 P_1，确定满足迪尔沃斯定理(定理 4.7)的"j"条链.
18. 对于图 4.24 中的严格偏序 P_2，确定一个满足迪尔沃斯定理(定理 4.6)的大小为"j"的反链.

272 19. 证明定理 4.7.
20. 证明：如果 V 是闭区间的任意集合，且存在一条从$[a, b]$到$[c, d]$的弧当且仅当 $a>d$，那么其结果有向图是一个严格偏序.
21. (a)证明：练习 20 所定义的有向图，满足定理 4.9 的必要条件.
(b)使用定理 4.9 确定图 4.19 到图 4.25 中的有向图中哪些是区间顺序?
22. (Trotter[1992])一个严格偏序是*平面的*(planar)，如果它有没有交叉边的图表示. [尽管存在任意大维数的平面严格偏序，Trotter[1992]还引用了 Baker 证明的一个结果，即一个格(如 4.4 节的定义)是平面的当且仅当它的维数至多等于 2.]
(a)通过重新画出没有交叉边的图表示，证明图 4.24 的严格偏序 P_1 是平面严格偏序.
(b)与图不同，平面严格偏序的子顺序不一定是平面的. 确定图 4.25 中的平面严格偏序 P 的一个非平面子顺序.

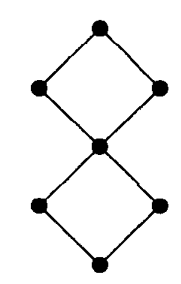


图 4.25 一个平面严格偏序

23. 回想 2.19.3 节中定义的拉姆齐数 $R(a, b)$并回顾 3.8 节. 证明：如果一个严格偏序至少有 $R(a+1, b+1)$个顶点，那么它或者有一条有 $a+1$ 个顶点的路径，或者有一个有 $b+1$ 个顶点的集合，且其中的任意两个顶点间都没有弧. (Dilworth[1950]的一个著名定理指出，只要这一严格偏序至少有 $ab+1$ 个顶点，那么相同的结论成立.)

273

4.4 格和布尔代数

4.4.1 格

设(X, R)是严格偏序. 在整个这一节我们都使用 R^* 表示由 $aR^*b\Leftrightarrow a=b$ 或 aRb 定义的 X 上的二元关系. 子集 $U\subseteq X$ 的*上界*(upper bound)是一个元素 $a\in X$，使得对于所有 $x\in U$ 有 aR^*x.

如果 a 是 U 的上界，且对于 U 的所有其他上界 b 有 aRb，那么称 a 为 U 的上确界(least upper bound)，即 lub U. 同时也使用术语 sup U，$\vee U$，U 的并(join of U). 类似地，我们定义子集 $U\subseteq X$ 的下界(lower bound)为使得对于所有 $x\in U$ 有 xR^*a 的元素 $a\in X$. 如果 a 是 U 的下界，且对于 U 的其他所有下界 b 有 aRb，那么 a 称为 U 的下确界(greatest lower bound)，即 glb U. 同时还使用术语 inf$U$，$\wedge U$，U 的交(meet of U). 如果 U 只是一对元素，比如说是$\{x, y\}$，a 是 U 的下确界，b 是 U 的上确界，我们可以写

$$a=x\wedge y,\quad b=x\vee y.$$

考虑图 4.24 中的严格偏序 P_2. 集合$\{y, w\}$没有上确界，集合$\{z, v\}$没有下确界. 然而，$\{z, v, y\}$有一个上确界，即 u，而$\{y, w\}$的下确界是 x. 注意 u 还是$\{u, z, v\}$的上确界.

【例 4.12 字典序(再探例 4.7)(Stanat McAllister[1977])】 在例 4.7 中，我们介绍了字典序(X, S). 考虑 Σ 是字母表$\{a, b\}$且 R 是由 $R=\{(a, b)\}$定义的严格线性序的情况. 于是，有 m 个 $a(m\geqslant 0)$的形如 $a^mb=aa\cdots ab$ 的串的集合没有上确界. 这是因为只有由 m 个 $a(m\geqslant 0)$组成的形如 $a^m=aa\cdots a$ 的串是上界，而且对于任意的 $n>m$，a^mSa^n. ■

格(lattice)是一个严格偏序，其中每一对元素都有一个交和一个并. 有时候，格还定义为严格偏序，其中每一个非空的元素子集有下确界和上确界. 当所考虑的是有限集合上的严格偏序时，这些定义是等价的(为什么？参见练习 4). 在下文中我们只考虑有限格，所以两个定义都是可接受的.

一些格的例子是：

- 给定集合的子集及其上的严格包含关系$\subsetneq$.
- 给定整数的因子集合及其上的“真因子”关系.
- 给定集合上的传递二元关系集合及其上的严格包含关系$\subsetneq$.
- 给定优先选择顺序集合的稳定匹配及其上的“支配”关系.

图 4.23 给出更多的例子.

一个格有多少个极大元素呢？考虑格(X, R)的所有极大元素，记为 $\max(X, R)$. 于是，lub $\max(X, R)$存在，且它不小于 $\max(X, R)$中的所有元素. 因此 $|\max(X, R)|=1$. 对于(X, R)的极小元素也可以运用类似的论证. 因此，我们有下面的定理. [274]

定理 4.11 *每一个格有一个最大元素和一个最小元素.*

同前一样，我们使用$\hat{1}$表示一个格的最大元素，使用$\hat{0}$表示一个格的最小元素. 在图 4.23的格中，给出了$\hat{1}$和$\hat{0}$.

下面是一组格的一些基本性质，其证明留作练习. 考虑格(X, R). 对于所有 $a, b, c, d\in X$：

- 如果 aRb 且 cRd，那么有

  $(a\wedge c)R(b\wedge d)$ 且 $(a\vee c)R(b\vee d)$ (保序性)
- $a\vee b=b\vee a$ 及 $a\wedge b=b\wedge a$ (交换性)
- $(a\vee b)\vee c=a\vee(b\vee c)$及$(a\wedge b)\wedge c=a\wedge(b\wedge c)$ (结合性)
- $a\wedge(a\vee b)=a\vee(a\wedge b)=a$ (吸收性)
- $a\vee a=a\wedge a=a$ (幂等性)

注意，分配性不在这一列表中. 其原因是，不是所有格都具有这一性质. 有意义的两个分配性质是：

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c) \tag{4.10}$$

及

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c). \tag{4.11}$$

对于格，条件(4.10)式和(4.11)式是等价的(参见练习9)．满足这两个条件的任意一个的格称为**分配格**(distributive lattice)．例如，在图4.26中，$z\wedge(x\vee y)=z$，而$(z\wedge x)\vee(z\wedge y)=\hat{0}$．因此，这个格不是分配格，因为条件(4.10)式不成立．

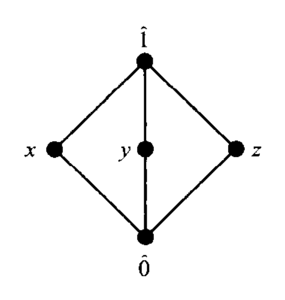


图4.26　一个非分配的补格

格可能有也可能没有的另外一个性质称为有补的．有最大元素$\hat{1}$和最小元素$\hat{0}$的格的一个元素x有补(complement)c，如果：

275

$$x \vee c = \hat{1} \text{ 且 } x \wedge c = \hat{0}.$$

例如，在图4.26中，x有补y和z．因为在任意格中，$\hat{0}$和$\hat{1}$总是相互为补的，这是该格中每一个元素都有补的情况．如果一个格的每一个元素都有补，那么这个格称为**补格**(complemented lattice)．既有补又是分配的格称为**布尔代数**(Boolean algebra)．格理论的这一领域(即布尔代数)有诸多重要的应用，例如在电子开关回路理论中的应用．参见Gregg[1998]、Greenlaw和Hoover[1998]．下面我们讨论布尔代数．

4.4.2　布尔代数

我们已经看到，一个元素x可以有多个补．然而，在布尔代数中这是不可能发生的．

定理4.12　*在布尔代数中，每个元素有一个且仅有一个补．*

证明　假设(X, R)是布尔代数且$x\in X$．假设x有两个不同的补y和z，于是利用分配性可以得到矛盾．详细的证明留作练习(练习15)．■

如果(X, R)是布尔代数且$x\in X$，我们使用x'表示x的补．

【例4.13 {0, 1}布尔代数】　设$X=\{0, 1\}$且在X上定义$R=\{(0, 1)\}$．于是(X, R)定义一个有下面性质的格

$$0 \vee 0 = 0, 0 \vee 1 = 1 \vee 0 = 1 \vee 1 = 1, 0 \wedge 0 = 0 \wedge 1 = 1 \wedge 0 = 0, 1 \wedge 1 = 1. \tag{4.12}$$

我们可以用下面的表概括(4.12)式：

∨	0	1
0	0	1
1	1	1

∧	0	1
0	0	0
1	0	1

．　(4.13)

我们有$\hat{1}=1$，$\hat{0}=0$，且1的补是0，0的补是1，这表明(X, R)是补格．我们可以通过下面的表概括这一结果：

′	
0	1
1	0

．　(4.14)

分配性的证明很容易，留给读者(练习11)．因此，(X, R)定义一个布尔代数．■

276

【例4.14 真值表】　把0考虑成表示一个陈述为假(F)且把1考虑成表示一个陈述为真(T)．我们可以把∨考虑成析取“或”，而把∧考虑成合取“且”．我们可以用下面的**真值表**(truth table)取代(4.13)式中的表：

或	F	T
F	F	T
T	T	T

且	F	T
F	F	F
T	F	T

(4.15)

第一个表对应于陈述“p 或 q”(有时候写成 $p \vee q$)为真当且仅当或者 p 为真，或者 q 为真，或者二者都为真；而第二个表对应于陈述“p 且 p”(有时候写成 $p \wedge q$)为真当且仅当 p 和 q 都为真. 例如，使用这些真值表，我们可以得出下列陈述为真的结论：

- 65>23 或华盛顿 DC 是美国的首都.
- 35+29=64 和蓝球队只有 5 名球员在场上打球.

然而，陈述

$$2+3=6 \quad \text{或} \quad 13\text{英寸}=1\text{英尺}$$

为假. 如果补′对应于否定“非”，表(4.14)式可以写成：

非	
F	T
T	F

(4.16)

使用(4.15)式和(4.16)式，我们可以分析给出的复杂陈述的情况. 这一分析将采用更大的真值表. 例如，考虑陈述“(p 或 q)且 p”[有时候写成“$(p \vee q) \wedge p$)”]. 我们可以使用下面的真值表分析这一陈述.

p	q	p 或 q	(p 或 q)且 p
F	F	F	F
F	T	T	F
T	F	T	T
T	T	T	T

前两列给出陈述 p 和 q 为 F 和 T 的所有组合. 第三列给出“p 或 q”在下列各情况下为真：p 和 q 分别是 F 和 T，或 T 和 F，或 T 和 T，就如(4.15)式的第一个表所给出的那样. 现在，在后两种情况下，“p 或 p”和 p 都是 T，而根据(4.15)式的第二个表，这使得“(p 或 q)且 p”为 T. 对于更复杂的陈述也可以使用类似的分析. 例如，考虑陈述“John 说谎且(Mary 说谎或 John 说真话).”令 p=“Jonh 说谎”且 q=“Mary 说谎”. 我们的陈述的真值表分析给出下面的表：

p	q	p'	q 或 p'	p 且(q 或 p')
F	F	T	T	F
F	T	T	T	F
T	F	F	F	F
T	T	F	T	T

277

这表明只有在 John 和 Mary 都说谎的情况下我们的陈述为真.

最后，考虑陈述“p'或 q”. 这一陈述的真值表由下表给出：

p	q	p'或 q
F	F	T
F	T	T
T	F	F
T	T	T

(4.17)

这一真值表也描述了条件(conditional)陈述"如果 p 则 q"的逻辑意义. 因为，当 p 为真时，"如果 p 则 q"只有当 p 也为真时才为真. 当 p 为假时，"如果 p 则 q"一定为真，因为这一陈述的"if"部分为假. 因为条件陈述经常出现在不同的上下文中，所以为表记方便，我们使用"$p\to q$"取代"p' 或 q"，即"$p'\vee q$". ■

【例 4.15 逻辑回路】 例 4.13 的布尔代数在计算机科学中非常重要. 我们可以把电网想象成是由带有两种电压"高"(1)或"低"(0)的电线设计而成的(另外，我们也可以考虑或者为"开"或者为"关"的开关). 我们可以考虑带有某种"门"的组合输入. 一个或门(or-gate)取两种电压 x 和 y 作为输入，而以电压 $x\vee y$ 作为输出，而一个与门(and-gate)取 x 和 y 作为输入，而以 $x\wedge y$ 作为输出，其中 $\vee$ 和 $\wedge$ 由(4.13)式定义. 例如，一个与门把两个高电压变成一个高电压，并把一个高电压和一个低电压变成一个低电压. 一个非门(inverter)接收电压 x 作为输入，以电压 x' 作为它的输出，即它把高电压变成低电压，并把低电压变成高电压.

图 4.27 给出图解或门、图解与门和图解非门. 图 4.28 给出一个回路图. 这里，我们接收三个输入 x_1，x_2，x_3 并计算输出 $(x_1\vee x_2)\wedge(x_2'\wedge x_3)$. 我们可以认为对应于这一回路图的计算机是计算下面的开关函数：

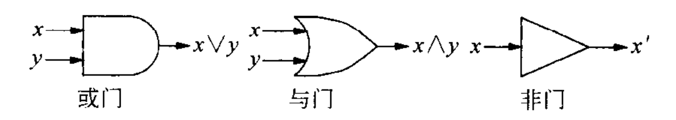


图 4.27　图解或门、图解与门和图解非门

$$f(x_1,x_2,x_3)=(x_1\vee x_2)\wedge(x'_2\wedge x_3).$$

利用(4.13)式和(4.14)式，我们如下计算 f：

x_1	x_2	x_3	$(x_1\vee x_2)\wedge(x_2'\wedge x_3)$
1	1	1	0
1	1	0	0
1	0	1	1
1	0	0	0
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	0

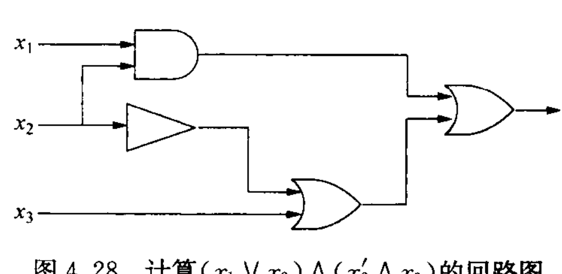


图 4.28　计算 $(x_1\vee x_2)\wedge(x_2'\wedge x_3)$ 的回路图

278

■

【例 4.16 顶灯】 考虑有一盏顶灯的房间. 进入这一房间的三个门口的每一个都有一个控制这一顶灯的开关. 一旦改变其中的一个开关，那么如果这一顶灯之前是开着的它就熄灭，如果这一顶灯之前是关着的它就打开. 这些开关是如何用电线连接起来的呢? 在大多数情况下，灯和开关是由一种称为与或回路(and-or circuit)的电路连接起来的. 与或回路是逻辑回路.

这一房间里的开关对应于这一与或回路的输入，而这一顶灯对应于该回路的唯一输出(也可以表示对应于多个顶灯的多个输出，但是，我们只考虑单输出的回路). 当该回路的输出是 1 时，这一顶灯是开着的，当该回路的输出是 0 时，这一顶灯是关着的. 假设我们房间里的开关被记为 x，y 和 z. 考虑输出为 $(x\wedge y\wedge z)\vee(x\wedge y'\wedge z')\vee(x'\wedge y\wedge z')\vee(x'\wedge y'\wedge z)$ 的一个与或回路. 所有可能的开关电压和其对应的顶灯结果概括如下：

x	y	z	$(x\wedge y\wedge z)\vee(x\wedge y'\wedge z')\vee(x'\wedge y\wedge z')\vee(x'\wedge y'\wedge z)$
1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	1
0	0	0	0

(4.18)

选出(4.18)式中的任意一行. 很容易检查 x，y 或 z 的一个变化将会产生由高电压变成低电压或低电压变成高电压的输出. 因此，这一与或门回路正好是我们的顶灯例子所需要的. 279

本节练习

1. 考虑图 4.24 中的严格偏序.
   (a)每一对元素都有上确界吗?
   (b)每一对元素都有下确界吗?
2. 在图 4.23 的每一个格中，确定：
   (a)lub$\{a, b, x\}$　　(b)glb$\{a, b, x\}$
   (c)$x\vee y$　　(d)$a\wedge b$
3. (a)图 4.20 中的哪些严格偏序是格?
   (b)图 4.21 中的哪些严格偏序是格?
4. 假设(X, R)是有限集合 X 上的严格偏序. 证明：X 的每一对元素有一个下确界和一个上确界，当且仅当 X 的每一个非空子集有一个下确界和一个上确界.
5. 对于图 4.20 中的每一个严格偏序，确定它是否有：
   (a)保序性　　(b)交换性　　(c)结合性
   (d)吸收性　　(e)幂等性　　(f)分配性
6. 对于图 4.21 中的每一个严格偏序重复练习 5.
7. 设 X 是任意集合，$\triangledown$和$\triangle$是二元操作，使得
   - $\triangledown$和$\triangle$是可交换的.
   - $\triangledown$和$\triangle$是结合的.
   - $\triangledown$和$\triangle$有吸收性.
   - $x\triangledown x=x\triangle x=x$

   证明(X, R)是严格偏序，其中对于所有的 x，$y\in X$，如果 $x\triangledown y=y$ 则 xRy.
8. 证明每一个格有：
   (a)交换性　　(b)结合性
   (c)吸收性　　(d)幂等性
9. 对于格，证明条件(4.10)式和(4.11)式是等价的.
10. 假设(X, R)是格且 aRb 及 cRd.
   (a)证明$(a\wedge c)R(b\wedge d)$.　　(b)证明$(a\vee c)R(b\vee d)$.
11. 设(X, R)是例 4.13 中的格. 证明(X, R)有分配性.
12. 确定图 4.23 中的每一格是否有补.
13. 确定图 4.23 中的每一个格是否是可分配的.

14. 假设(X, R)是分配格. 证明：如果yRx，则对于所有$z\in X$，$y\vee(x\wedge z)=x\wedge(y\vee z)$（带有此性质的格称为模格(modular)）.

280

15. 证明：在一个布尔代数中，每一个元素都有唯一的补. [提示：使用分配性证明每一个元素的补一定是唯一的.]
16. 构建下列陈述的真值表：

    (a)$p\wedge q'$ (b)$(p\wedge q)\vee(p'\wedge q')$

    (c)$(p\wedge q')\to q$ (d)$(p\wedge q)\to(p\wedge r)$
17. 确定下列陈述的符号形式，然后构建真值表：

    (a)如果 Pete 爱 Christine，那么 Christine 爱 Pete.

    (b)Pete 和 Christine 彼此相爱.

    (c)Pete 爱 Christine 而 Christine 不爱 Pete 不是真的.
18. 说两个陈述是等价的(equivalent)，如果一个为真当且仅当另一个为真. 你可以通过构建两个陈述的真值表，并证明在相应行有 T 来说明它们是等价的. 使用这一想法检查下列各对陈述是否等价：

    (a)$p'\wedge q'$且$(p\vee q)'$ (b)$p\vee(q\wedge r)$且$(p\vee q)\wedge(p\vee r)$

    (c)$p\vee(p\wedge q)$且 p (d)$(p\to q)\to r$ 且 $p\to(q\to r)$

    (e)$(p\vee q)\to(p\wedge q)$且$((p\vee q)\wedge(p\wedge q)')\to(p\vee q)'$
19. 考虑条件陈述 $p\to q$ 及其相关陈述：$q\to p$(逆)，$p'\to q'$(反)，$q'\to p'$(逆反). 这四个陈述中哪对陈述是等价的？(参见练习 18.)
20. 给出下列逻辑回路的开关函数：

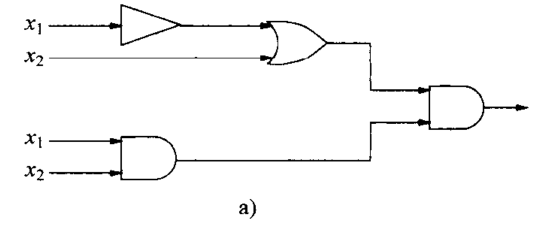
281

    (a)图 4.29a (b)图 4.29b
21. 画出下列开关函数的逻辑回路：

    (a)$(p\wedge q)\vee(p'\vee q)'$

    (b)$(p\vee(q\wedge r))\wedge((p\vee q)\wedge(p\vee r))$

    (c)$(p\to q)\vee q$[参见(4.17)式]
22. 寻找一个与或回路来模型化有两个开关的一个顶灯. [提示：考虑陈述 $A\vee B$，其中 A 和 B 是下面的每一个陈述：$p\wedge q$，$p'\wedge q$，$p\wedge q'$，$p'\wedge q'$.]



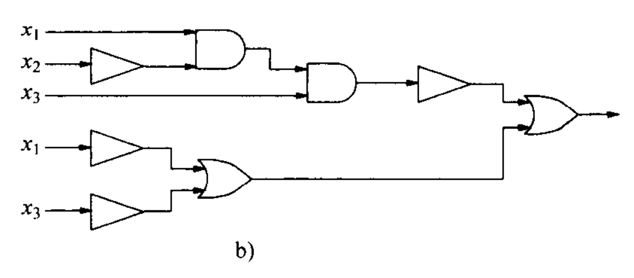


图 4.29 两个逻辑回路

参考文献

BAKER, K. A., FISHBURN, P. C., and ROBERTS, F. S., "Partial Orders of Dimension 2, Interval Orders and Interval Graphs," *Networks, 2* (1972), 11–28.

BARBERÀ, S., HAMMOND, P. J., and SEIDEL, C. (eds.), *Handbook of Utility Theory*, Vol. 1, Kluwer Academic Publishers, Boston, 2004.

BENTHAM, J., *The Principles of Morals and Legislation*, London, 1789. (Available online at: http://www.la.utexas.edu/research/poltheory/bentham/ipml/)

BOGART, K. P., RABINOVITCH, I., and TROTTER, W. T., "A Bound on the Dimension of Interval Orders," *J. Comb. Theory, 21* (1976), 319–328.

DILWORTH, R. P., "A Decomposition Theorem for Partially Ordered Sets," *Ann. Math., 51* (1950), 161–166.

DUSHNIK, B., and MILLER, E. W., "Partially Ordered Sets," *Amer. J. Math., 63* (1941), 600–610.

FALMAGNE, J.-C., *Elements of Psychophysical Theory*, Oxford University Press, New York, 1985.

FENTON, N. E., and PFLEEGER, S. L., *Software Metrics*, 2nd ed., PWS Publishing Co., Boston, 1997.

FISHBURN, P. C., "Intransitive Indifference with Unequal Indifference Intervals," *J. Math. Psychol., 7* (1970), 144–149. (a)

FISHBURN, P. C., *Utility Theory for Decisionmaking*, Wiley, New York, 1970. (b)

GALE, D., and SHAPLEY, L. S., "College Admissions and the Stability of Marriage," *Amer. Math. Monthly, 69* (1962), 9–15.

GREENLAW, R., and HOOVER, J., *Fundamentals of the Theory of Computation: Principles and Practice*, Morgan Kaufmann Publishers, San Francisco, 1998.

GREGG, J., *Ones and Zeros: Understanding Boolean Algebra, Digital Circuits and the Logic of Sets*, IEEE, Piscataway, NJ, 1998.

GUSFIELD, D., *Algorithms on Strings, Trees and Sequences; Computer Science and Computational Biology*, Cambridge University Press, New York, 1997.

GUSFIELD, D., and IRVING, R. W., *The Stable Marriage Problem: Structure and Algorithms*, MIT Press, Cambridge, MA, 1989.

HIRAGUCHI, T., "On the Dimension of Orders," *Sci. Rep. Kanazawa Univ., 4* (1955), 1–20.

KEENEY, R. L., and RAIFFA, H., *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, Cambridge University Press, New York, 1993.

KENDALL, M. G., "A Statistical Approach to Flinders Petrie's Sequence Dating," *Bull. Intern. Statist. Inst., 40* (1963), 657–680.

KENDALL, M. G., "Incidence Matrices, Interval Graphs, and Seriation in Archaeology," *Pacific J. Math., 28* (1969), 565–570. (a)

282

KENDALL, M. G., "Some Problems and Methods in Statistical Archaeology," *World Archaeology, 1* (1969), 61–76. (b)

KOMM, H., "On the Dimension of Partially Ordered Sets," *Amer. J. Math., 70* (1948), 507–520.

LUCE, R. D., *Utility of Gains and Losses*, Lawrence Erlbaum Associates, Mahwah, NJ, 2000.

PETRIE, W. M. F., "Sequences in Prehistoric Remains," *J. Anthrop. Inst., N.S., 29* (1899), 295–301.

PETRIE, W. M. F., *Diospolis Parra*, Egypt Exploration Fund, London, 1901.

ROBERTS, F. S., *Discrete Mathematical Models, with Applications to Social, Biological, and Environmental Problems*, Prentice Hall, Englewood Cliffs, NJ, 1976.

ROBERTS, F. S., "Indifference and Seriation," *Ann. N.Y. Acad. Sci., 328* (1979), 173–182. (a)

ROBERTS, F. S., *Measurement Theory, with Applications to Decisionmaking, Utility, and the Social Sciences*, Addison-Wesley, Reading, MA, 1979. (b)

SPERNER, E., "Ein Satz über Untermengen einer endlichen Menge," *Math. Zeit., 27* (1928), 544–548.

STANAT, D. F., and MCALLISTER, D. F., *Discrete Mathematics in Computer Science*, Prentice Hall, Englewood Cliffs, NJ, 1977.

SZPILRAJN, E., "Sur l'Extension de l'Ordre Partiel," *Fund. Math., 16* (1930), 386–389.

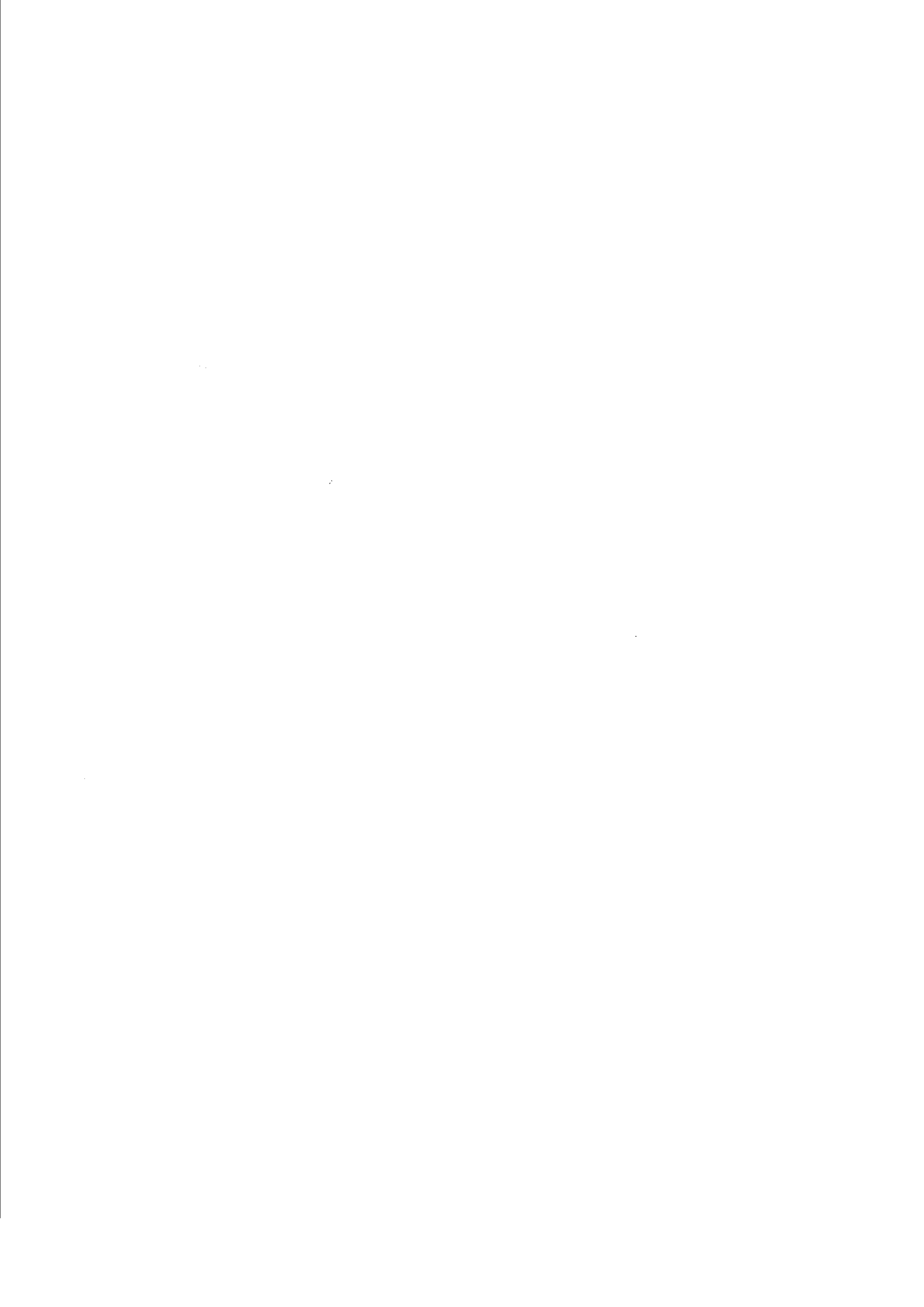
TROTTER, W. T., *Combinatorics and Partially Ordered Sets*, The Johns Hopkins University Press, Baltimore, MD, 1992.

TROTTER, W. T., "Graphs and Partially Ordered Sets: Recent Results and New Directions," *Congressus Num., 116* (1996), 253–278.

VINCKE, P., *Multicriteria Decision-Aid*, Wiley, Chichester, UK, 1992.

YANNAKAKIS, M., "The Complexity of the Partial Order Dimension Problem," *SIAM J. Algebraic Discrete Methods, 3* (1982), 351–358.

283



第二部分　计数问题

第5章　生成函数及其应用[⊖]

5.1　生成函数的例子

组合数学的很多内容都是研究计数工具的开发．我们已经看到计数排列数或模式数量通常很重要，但是实践中，不可能列出所有排列．因此，我们需要帮助计数的工具．在以下四章中，我们给出一系列用于计数的工具．我们给出的最强有力的工具之一是生成函数的概念．本章重点研究生成函数．

在组合数学中，我们通常设法计数依赖于输入，或参数 k 的量 a_k．例如，如果输入的大小为 k，那么实施一个计算所需要的步数 a_k 就是如此．我们通过讨论未知值 a_0，a_1，a_2，…，a_k，…的序列来形式化这一关于 k 的相关性．我们设法确定在这一序列中的第 k 项．生成函数给出了编码诸如 a_0，a_1，a_2，…，a_k，…这样的序列的简单方法，它可以容易被"解码"以发现这一序列中的各项．我们将会看到该技巧在没有这一序列的情况下如何计算这一序列的编码函数，或生成函数．于是，可以解码找到 a_k．这一方法使得我们能够以间接但高效的方式确定未知量 a_k．

我们将给出的生成函数方法是一个古老方法．它源于 De Moivre 在 1720 前后的工作，1748 年欧拉在研究关于划分的问题中发展了这一方法，拉普拉期于 18 世纪末期及 19 世纪初期对其进行了广泛的论述，其探究主要与概率论相关．尽管这一方法有其悠久的历史，但是正如我们将看到的那样，这一方法仍有着广泛的应用．关于生成函数的更全面讨论可以参见 Lando[2003]、MacMahon[1960]、Riordan[1980]、Srivastava 和 Manocha[1984]、Wilf[1994]（还可以参见 Riordan[1964]）．

285

5.1.1　幂级数

本章中，我们使用来自于微积分的一个基础概念，幂级数的概念．我们所需要的关于幂级数的结果全都概括于本小节中．想要了解更多细节的读者，包括这些结果的证明，可以参考大部分微积分的书籍．

幂级数（power series）是形如 $\sum_{k=0}^{\infty} a_k x^k$ 的无穷级数．这样的无穷级数对于 $x=0$ 总是收敛的．对于 x 的其他值，它或者不收敛，或者存在一个正整数 R（可能是无穷的）使得对于所有满足 $|x|<R$ 的 x，它收敛．在后一种情况中，这样的最大 R 称为**收敛半径**（radius of convergence）．对于前面的情况，我们说 0 是收敛半径．可以把幂级数 $\sum_{k=0}^{\infty} a_k x^k$ 考虑成 x 的一个函数 $f(x)$，这一函数对无穷和收敛的那些 x 值有定义，而且通过计算这一无穷和来计算这个函数．本章的大部

⊖ 在初级课程中，本章可以跳过．只有第 5 章和第 6 章需要微积分的知识，其他只假设来自于微积分课程的一定程度的数学经历．

分内容不再谈及收敛的问题．我们只简单地假设已选定了使得 $\sum_{k=0}^{\infty} a_k x^k$ 收敛的 x[⊖]．

幂级数以下面的形式出现在微积分中．假设 $f(x)$ 是一个函数，它对于一个包含 0 的区间内的所有 x 有任意阶的导数．那么

$$f(x)=\sum_{k=0}^{\infty}\frac{f^{(k)}(0)}{k!}x^k=f(0)+f'(0)x+\frac{f''(0)}{2!}x^2+\frac{f'''(0)}{3!}x^3+\cdots. \tag{5.1}$$

(5.1)式右边的幂级数对于 x 的某些值收敛，至少对于 $x=0$ 收敛．这一幂级数称为 f 的*麦克劳林展开*(Maclaurin expansion)，或称为 f 关于 $x=0$ 的*泰勒级数展开*(Taylor series expansion)．

下面是若干最著名和最有用的麦克劳林展开：

$$\frac{1}{1-x}=\sum_{k=0}^{\infty}x^k=1+x+x^2+x^3+\cdots,\text{对于 } |x|<1, \tag{5.2}$$

$$e^x=\sum_{k=0}^{\infty}\frac{1}{k}x^k=1+x+\frac{1}{2!}x^2+\frac{1}{3!}x^3+\cdots,\text{对于 } |x|<\infty, \tag{5.3}$$

$$\sin x=\sum_{k=0}^{\infty}\frac{(-1)^k}{(2k+1)!}x^{2k+1}=x-\frac{1}{3!}x^3+\frac{1}{5!}x^5-\cdots,\text{对于 } |x|<\infty, \tag{5.4}$$

286

$$\ln(1+x)=\sum_{k=1}^{\infty}\frac{(-1)^{k+1}}{k}x^k=x-\frac{1}{2}x^2+\frac{1}{3}x^3-\frac{1}{4}x^4+\cdots,\text{对于 } |x|<1. \tag{5.5}$$

例如，为了说明(5.3)式是(5.1)式的特殊情况，只要观察如果 $f(x)=e^x$ 那么对于所有的 k 有 $f^{(k)}(x)=e^x$ 以及 $f^{(k)}(0)=1$ 即可．读者应该核实在相同的条件下(5.2)式，(5.3)式和(5.5)式也是(5.1)式的特殊情况．

幂级数之所以有用的原因之一就是很容易对它们做相加、相乘、相除、合成、微分和积分运算．我们通过公式化几个一般的原理来使读者记住幂级数的这些性质．

原理 1　假设 $f(x)=\sum_{k=0}^{\infty}a_kx^k$ 及 $g(x)=\sum_{k=0}^{\infty}b_kx^k$．那么分别通过逐项相加、外乘或使用长除法可以计算出 $f(x)+g(x)$，$f(x)g(x)$ 和 $f(x)/g(x)$．[对于除法只有当所讨论的 x 的值 $g(x)$ 不等于零时成立．]特别地，有

$$\begin{aligned}
f(x)+g(x)&=\sum_{k=0}^{\infty}(a_k+b_k)x^k\\
&=(a_0+b_0)+(a_1+b_1)x+(a_2+b_2)x^2+\cdots,\\
f(x)g(x)&=a_0\sum_{k=0}^{\infty}b_kx^k+a_1x\sum_{k=0}^{\infty}b_kx^k+a_2x^2\sum_{k=0}^{\infty}b_kx^k+\cdots\\
&=a_0(b_0+b_1x+b_2x^2+\cdots)+a_1x(b_0+b_1x+b_2x^2+\cdots)\\
&\quad+a_2x^2(b_0+b_1x+b_2x^2+\cdots)+\cdots,\\
\frac{f(x)}{g(x)}&=\frac{a_0}{\sum_{k=0}^{\infty}b_kx^k}+\frac{a_1x}{\sum_{k=0}^{\infty}b_kx^k}+\frac{a_2x^2}{\sum_{k=0}^{\infty}b_kx^k}+\cdots\\
&=\frac{a_0}{b_0+b_1x+b_2x^2+\cdots}+\frac{a_1x}{b_0+b_1x+b_2x^2+\cdots}\\
&\quad+\frac{a_2x^2}{b_0+b_1x+b_2x^2+\cdots}+\cdots.
\end{aligned}$$

⊖ 通过把 $\sum_{k=0}^{\infty}a_kx^k$ 简单地考虑为一种形式表示，即*形式幂级数*(formal power serie)，而不是一个函数，而且通过对这些形式表示进行适当的处理会使得这一假设更加精确．关于这一方法的详情，参见 Niven[1969]．

如果对于 $|x|<R$，幂级数 $f(x)$ 和 $g(x)$ 都收敛，那么 $f(x)+g(x)$ 和 $f(x)g(x)$ 也收敛. 如果 $g(0)\neq 0$，那么在 0 周围的某个区间内 $f(x)/g(x)$ 也收敛.

例如，利用(5.2)式和(5.3)式，我们有

$$
\begin{aligned}
\frac{1}{1-x}+e^x &= (1+x+x^2+x^3+\cdots)+\left(1+x+\frac{1}{2!}x^2+\frac{1}{3!}x^3+\cdots\right)\\
&= (1+1)+(1+1)x+\left(1+\frac{1}{2!}\right)x^2+\left(1+\frac{1}{3!}\right)x^3+\cdots\\
&= \sum_{k=0}^{\infty}\left(1+\frac{1}{k!}\right)x^k.
\end{aligned}
$$

也有

$$
\begin{aligned}
\frac{1}{1-x}e^x &= (1+x+x^2+x^3+\cdots)\left(1+x+\frac{1}{2!}x^2+\frac{1}{3!}x^3+\cdots\right)\\
&= 1\left(1+x+\frac{1}{2!}x^2+\frac{1}{3!}x^3+\cdots\right)+x\left(1+x+\frac{1}{2!}x^2+\frac{1}{3!}x^3+\cdots\right)\\
&\quad +x^2\left(1+x+\frac{1}{2!}x^2+\frac{1}{3!}x^3+\cdots\right)+\cdots\\
&= 1+2x+\frac{5}{2}x^2+\frac{5}{3}x^3+\cdots.
\end{aligned}
$$

287

在函数合成下，幂级数也很容易计算.

原理 2 如果 $f(x)=g(u(x))$ 并且我们知道 $g(u)=\sum_{k=0}^{\infty}a_k u^k$，那么我们有 $f(x)=\sum_{k=0}^{\infty}a_k[u(x)]^k$ ㊀.

因此，在等式(5.5)中令 $u=x^4$，我们得

$$\ln(1+x^4)=\sum_{k=1}^{\infty}\frac{(-1)^{k+1}}{k}x^{4k}.$$

[原理 2 扩展到关于 $u(x)$ 的幂级数的情况.]

原理 3 如果幂级数 $f(x)=\sum_{k=0}^{\infty}a_k x^k$ 对于所有 $|x|<R$ 收敛，且 $R>0$，那么可以通过逐项微分和积分计算 $f(x)$ 的导数和不定积分. 即

$$\frac{df}{dx}(x)=\frac{d}{dx}\left(\sum_{k=0}^{\infty}a_k x^k\right)=\sum_{k=0}^{\infty}\frac{d}{dx}(a_k x^k)=\sum_{k=0}^{\infty}ka_k x^{k-1} \tag{5.6}$$

及

$$\int_0^x f(t)dt=\int_0^x\left(\sum_{k=0}^{\infty}a_k t^k\right)dt=\sum_{k=0}^{\infty}\int_0^x a_k t^k dt=\sum_{k=0}^{\infty}\frac{1}{k+1}a_k x^{k+1}. \tag{5.7}$$

对于 $|x|<R$，(5.6)式和(5.7)式中的幂级数也收敛.

例如，因为

$$\frac{1}{(1-x)^2}=\frac{d}{dx}\left[\frac{1}{1-x}\right],$$

所以我们从(5.2)式和(5.6)式看到：

$$\frac{1}{(1-x)^2}=\sum_{k=0}^{\infty}kx^{k-1}=1+2x+3x^2+4x^3+\cdots.$$

㊀ 如果对于 $|u|<S$ 幂级数 $g(u)$ 收敛，且只要 $|x|<R$ $|u(x)|<S$，那么对于所有 $|x|<R$，幂级数 $f(x)$ 收敛.

5.1.2　生成函数

假设我们对计算数字序列(a_k)的第k项感兴趣．习惯上，我们使用(a_k)表示序列，用不带括号的a_k表示第k项．序列(a_k)的(普通)生成函数((ordinary)generating function)定义如下：

288

$$G(x)=\sum_{k=0}^{\infty}a_kx^k=a_0x^0+a_1x^1+a_2x^2+\cdots. \tag{5.8}$$

如果这一序列是有限的则和是有限的，如果这一序列是无穷的则和是无穷的．在后一种情况下，我们认为x是使得(5.8)式中的和收敛的值.

【例 5.1】　假设对于$k=0$，1，…，n，$a_k=\binom{n}{k}$．那么序列(a_k)的普通生成函数是

$$G(x)=\binom{n}{0}+\binom{n}{1}x+\binom{n}{2}x^2+\cdots+\binom{n}{n}x^n.$$

根据二项式展开(定理2.7)，有

$$G(x)=(1+x)^n.$$

我们所做的处理的好处就是用一种简单闭式(编码形式)表示了$G(x)$．知道了$G(x)$的这一简单形式，现在你有可能通过记忆$G(x)$的这一闭式并解码而得到a_k，即展开它并找寻x^k的系数．正如之前我们所看到的那样，更有用的事实是，我们经常能够在不知道a_k的情况下求出$G(x)$，然后通过展开求出a_k. ■

【例 5.2】　假设对于$k=0$，1，2，…，$a_k=1$．那么

$$G(x)=1+x+x^2+\cdots.$$

根据等式(5.2)，对于$|x|<1$有

$$G(x)=\frac{1}{1-x}.$$

同样读者将注意到这是$G(x)$的闭式. ■

【例 5.3】　我们经常只知道生成函数而不知道序列．我们尝试着从生成函数"恢复"这一序列．例如，假设

$$G(x)=\sum_{k=0}^{\infty}a_kx^k,$$

且我们知道

$$G(x)=\frac{x^2}{1-x}.$$

a_k是什么呢？利用(5.2)式，对于$|x|<1$，我们有

$$\begin{aligned}G(x)&=x^2\left[\frac{1}{1-x}\right]\\&=x^2(1+x+x^2+\cdots)\\&=x^2+x^3+x^4+\cdots.\end{aligned}$$

289

因此

$$(a_k)=(0,0,1,1,1,\cdots).$$

本章与第6章中，我们将研究各种展开$G(x)$的技术以得到所需序列(a_k). ■

【例 5.4】　假设对于$k=0$，1，2，…，$a_k=1/k!$．那么有

$$G(x)=\frac{1}{0!}+\frac{1}{1!}x+\frac{1}{2!}x^2+\frac{1}{3!}x^3+\cdots.$$

根据等式(5.3)，对于 x 的所有值，$G(x)=e^x$. ■

【例 5.5】 假设 $G(x)=x\sin(x^2)$是序列(a_k)的普通生成函数. 为了寻找 a_k，利用等式(5.4)，用 x^2 替换 x，并乘以 x，得到

$$\begin{aligned}G(x) &= x\left[x^2-\frac{1}{3!}(x^2)^3+\frac{1}{5!}(x^2)^5-\cdots\right]\\&= x^3-\frac{1}{3!}x^7+\frac{1}{5!}x^{11}-\cdots.\end{aligned}$$

因此，我们看到 a_k 是下面序列的第 k 项：

$$0,0,0,1,0,0,0,-\frac{1}{3!},0,0,0,\frac{1}{5!},0,\cdots.$$ ■

【例 5.6】 假设 $G(x)=\cos x$ 是序列(a_k)的普通生成函数. 因为 $G(x)$有任意阶的导数，所以通过对所有的 k 计算 $f^{(k)}(0)$，我们可以进行麦克劳林级数展开，并得到

$$G(x)=\sum_{k=0}^{\infty}\frac{(-1)^k}{(2k)!}x^{2k}=1-\frac{1}{2!}x^2+\frac{1}{4!}x^4-\cdots. \tag{5.9}$$

这一展开式的证实留作练习. 另外一个方法是观察 $G(x)=\mathrm{d}(\sin x)/\mathrm{d}x$，所以可以使用等式(5.4). 因此，我们看到

$$\begin{aligned}G(x) &= \frac{\mathrm{d}}{\mathrm{d}x}(\sin x)\\&= \frac{\mathrm{d}}{\mathrm{d}x}\left[x-\frac{1}{3!}x^3+\frac{1}{5!}x^5-\cdots\right]\\&= \frac{\mathrm{d}}{\mathrm{d}x}[x]+\frac{\mathrm{d}}{\mathrm{d}x}\left[-\frac{1}{3!}x^3\right]+\frac{\mathrm{d}}{\mathrm{d}x}\left[\frac{1}{5!}x^5\right]+\cdots\\&= 1-\frac{1}{2!}x^2+\frac{1}{4!}x^4-\cdots,\end{aligned}$$

上述展开式与(5.9)式一样. ■ [290]

【例 5.7】 如果$(a_k)=(1, 1, 1, 0, 1, 1, \cdots)$，则普通生成函数由下式给出：

$$\begin{aligned}G(x) &= 1+x+x^2+x^4+x^5+\cdots\\&= (1+x+x^2+x^3+x^4+x^5+\cdots)-x^3\\&= \frac{1}{1-x}-x^3.\end{aligned}$$ ■

【例 5.8】 如果$(a_k)=(1/2!, 1/3!, 1/4!, \cdots)$，则普通生成函数由下式给出：

$$\begin{aligned}G(x) &= \frac{1}{2!}+\frac{1}{3!}x+\frac{1}{4!}x^2+\cdots\\&= \frac{1}{x^2}\left(\frac{1}{2!}x^2+\frac{1}{3!}x^3+\frac{1}{4!}x^4+\cdots\right)\\&= \frac{1}{x^2}(e^x-1-x).\end{aligned}$$ ■

【例 5.9 标签图的数量】 在 3.1.3 节中，我们计数了有 n 个顶点和 e 条边且 $n\geqslant 2$，$e\leqslant C(n, 2)$ 的标签图的数量为 $L(n, e)$. 固定 n，且令 $a_k=L(n, k)$，$k=0, 1, \cdots, C(n, 2)$，考虑下面的生成函数：

$$G_n(x) = \sum_{k=0}^{C(n,2)} a_k x^k.$$

注意，在 3.1.3 节中，我们计算得到 $L(n, e)=C(C(n, 2), e)$. 因此，如果 $r=C(n, 2)$，那么

$$G_n(x) = \sum_{k=0}^{r} C(r,k) x^k.$$

根据二项式展开(定理 2.7)，我们有

$$G_n(x) = (1+x)^r = (1+x)^{C(n,2)} \tag{5.10}$$

这一表达式是概括数 $L(n, e)$的内容的一个简单方式. 特别地，从(5.10)式我们能够得到有 n 个顶点的标签图的数量 $L(n)$的公式. 因为

$$L(n) = \sum_{k=0}^{r} C(r,k),$$

它等于 $G_n(1)$. 因此，在(5.10)式中取 $x=1$，我们有

$$L(n) = 2^{C(n,2)},$$

291 这就是我们在 3.1.3 节中得到的结果. ■

【例 5.10 车多项式：工作分配和计算机程序存储】 假设 B 是诸如图 5.1 和图 5.2 所示的 $n\times m$ 的方格盘，其中有一些被禁止的方格，而其他则是可接受的方格，可接受的方格被涂成深色. 设 $r_k(B)$是选择满足下面条件的 k 个可接受(深色)方格的方法数量：任意两个可接受方格不在同一行上也不在同一列上. 我们可以把 B 看成是一个棋盘的一部分. 车(rook)是在这一棋盘上既能走水平线又能走垂线的棋子. 因此，一个车能够吃掉另一个车，如果这两个车在同一行上或在同一列上. 我们希望在 B 上的可接受方格中放置 k 个车，使得没有车能够吃掉另一个车. 因此，$r_k(B)$计数在 B 的可接受方格内放置 k 个不能吃掉的车的方法数量.

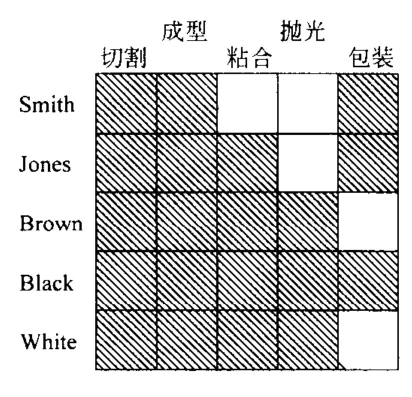


图 5.1 对应于工作分配问题的棋盘

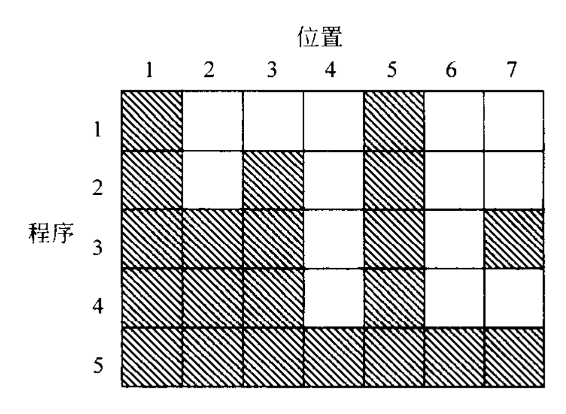


图 5.2 对应于计算机存储问题的棋盘

图 5.1 的 5×5 棋盘出现于一个工作分配问题. 行对应于工人，列对应于工作，位置 i，j 被涂成深色，如果工人 i 适合工作 j. 我们想要确定可以为每一个工人分配一个工作的方法数量，使得每一项工作只有一个工人，这样一个工人只能得到他适合的一项工作. 容易看到这等价于计算 $r_5(B)$的问题.

图 5.2 的 5×7 棋盘出现于计算机程序存储问题. 如果存储位置 j 对于程序 i 有足够的存储容量，那么位置 i，j 被涂成深色. 我们想要把每一个程序指定给一个有足够存储容量的存储位置，每一个位置至多有一个程序. 能够这样做的方法数量也由 $r_5(B)$给出. 在下文和 7.1 节的练习中，我们将计算这两个例子中的 $r_5(B)$.

表达式

$$R(x,B) = r_0(B) + r_1(B)x + r_2(B)x^2 + \cdots$$

称为棋盘 B 的车多项式(rook polynomial). 这一车多项式实际上是关于 x 的多项式，因为对于大于可接受方格数量的 k，$r_k(B)=0$.

292

这一车多项式正好是下面序列的普通生成函数：

$$(r_0(B), r_1(B), r_2(B), \cdots).$$

正如对一般的生成函数一样，我们将在非显示计算系数 $r_k(B)$ 的情况下寻找计算车多项式的方法，然后我们将可以根据这一多项式来计算这些系数.

作为例子，考虑图 5.3 的两个棋盘 B_1 和 B_2. 在棋盘 B_1 中，没有车时有一种放置方法(对于任意的棋盘都是如此)，放置一个车时有两种方法(使用其中一个深色方格)，放置两个车时有一种方法(使用两个深色方格)，没有放置大于两个车的方法. 因此

$$R(x,B_1) = 1 + 2x + x^2.$$

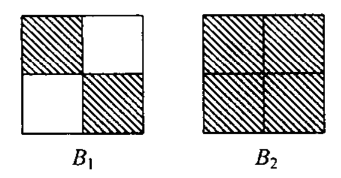


图 5.3　两个棋盘

在棋盘 B_2 中，同样也有存在一种放置 0 个车的方法，存在四种放置一个车的方法(使用任意一个深色方格)，存在两种放置两个车的方法(使用对角线上的方格或非对角线上的方格)，不存在放置大于两个车的方法. 因此

$$R(x,B_2) = 1 + 4x + 2x^2.$$ ■

本节练习

1. 对于下列每一个函数，通过计算导数 $f^{(k)}(0)$ 确定它的麦克劳林展开.

(a) $\cos x$　(b) e^{3x}　(c) $\sin(2x)$　(d) x^3+4x+7

(e) x^2+e^x　(f) xe^x　(g) $\ln(1+4x)$　(h) $x\sin x$

293

2. 对于下列每一个函数，使用已知的麦克劳林展开通过加、合成、微分等等方法确定其麦克劳林展开.

(a) $x^3+\frac{1}{1-x}$　(b) $x^2\frac{1}{1-x}$　(c) $\sin x^4$　(d) $\frac{1}{4-x}$

(e) $\sin(x^2+x+1)$　(f) $\frac{1}{(1-x)^3}$　(g) $5e^x+e^{3x}$　(h) $\ln(1-x)$

(i) $\ln(1+3x)$　(j) $x^3\sin x^5$　(k) $\ln(1+x^2)$　(l) $\frac{1}{1-2x}e^x$

3. 对于下列序列，确定其普通生成函数并做尽可能的简化.

(a) (1, 1, 1, 0, 0, …)　(b) (1, 0, 2, 3, 4, 0, 0, …)　(c) (5, 5, 5…)

(d) (1, 0, 0, 1, 1, …)　(e) (0, 0, 1, 1, 1, 1, 1, …)　(f) (0, 0, 4, 4, 4, …)

(g) (1, 1, 2, 1, 1, 1, …)　(h) $(a_k)=\left(\frac{3}{k!}\right)$　(i) $(a_k)=\left(\frac{3^k}{k!}\right)$

(j) $\left(0, 0, \frac{1}{2!}, \frac{1}{3!}, \frac{1}{4!}, \cdots\right)$　(k) $\left(1, 1, \frac{1}{2!}, \frac{1}{3!}, \frac{1}{4!}, \cdots\right)$　(l) $\left(\frac{1}{3!}, \frac{1}{4!}, \frac{1}{5!}, \cdots\right)$

(m) $\left(3, -\frac{3}{2}, \frac{3}{3}, -\frac{3}{4}, \cdots\right)$　(n) (1, 0, 1, 0, 1, 0, …)　(o) (0, 1, 0, 3, 0, 5, …)

(p) $\left(2, 0, -\frac{2}{3!}, 0, \frac{2}{5!}, \cdots\right)$　(q) $\left(1, -1, \frac{1}{2!}, -\frac{1}{3!}, \frac{1}{4!}, \cdots\right)$　(r) $\left(0, -1, \frac{1}{2}, -\frac{1}{3}, \frac{1}{4}, \cdots\right)$

4. 确定由下列普通生成函数给定的序列.

(a) $(x+5)^2$　(b) $(1+x)^4$　(c) $\frac{x^3}{1-x}$

(d) $\frac{1}{1-3x}$　(e) $\frac{1}{1+8x}$　(f) e^{6x}

(g) $1+\frac{1}{1-x}$　(h) $5+e^{2x}$　(i) $x\sin x$

(j) $x^3+x^4+e^{2x}$　　(k) $\frac{1}{1-x^2}$　　(l) $2x+e^{-x}$

(m) e^{-2x}　　(n) $\sin 3x$　　(o) $x^2\ln(1+2x)+e^x$

(p) $\frac{1}{1+x^2}$　　(q) $\cos 3x$　　(r) $\frac{1}{(1+x)^2}$

5. 假设序列(a_k)的普通生成函数是下面所给出的函数. 在每一种情况下，确定a_3.

(a) $(x-7)^3$　　(b) $\frac{14}{1-x}$　　(c) $\ln(1-2x)$　　(d) e^{5x}

294

6. 对于练习5的每一种情况，确定a_4.

7. Jones教授想要讲授微积分I或线性代数，Smith教授想要讲授线性代数或组合数学，Green教授想要讲授微积分I或组合数学. 可以至多为每一位教授指定一门课程，每门课程最多有一位教授，一位教授只能得到他想要讲授的一门课程. 建立一个生成函数并使用这一生成函数回答下列问题.

(a)我们有多少种方法给一位教授指定一门课程.

(b)我们有多少种方法给两位教授指定课程.

(c)我们有多少种方法给三位教授指定课程.

8. 假设工人a适合工作3，4，5，工人b适合工作2，3，而工人c适合工作1，5. 同样，每一名工人至少指定一项工作，每一项工作不超过一名工人，一名工人只得到他所适合的一项工作. 建立一个生成函数并使用它回答下列问题.

(a)我们有多少种方法给一名工人指定一项工作?

(b)我们有多少种方法给两名工人指定工作?

(c)我们有多少种方法给三名工人指定工作?

9. 假设T_n是n个顶点的(无标签)根树的数量. 普通生成函数$T(x)=\sum T_nx^n$由下式给出:

$$T(x)=x+x^2+2x^3+4x^4+9x^5+20x^6+\cdots.$$

(Riordan[1980]对于$n\leqslant 26$计算了T_n. 现在已知更多T_n的值.)证实x，x^2，x^3，x^4，x^5和x^6的系数.

10. 设$M(n,a)$是有n个顶点和a条弧的标签有向图的数量，并设$M(n)$是有n个顶点的标签有向图的数量(参见3.1.3节). 如果固定n，设$b_k=M(n,k)$，并设

$$D_n(x)=\sum_{k=0}^{n(n-1)} b_kx^k.$$

(a)确定$D_n(x)$的一个简单、闭式的表达式.

(b)利用这一表达式推导$M(n)$的一个公式.

11. 假设$c_k=R(n,k)$是有某个性质P且有n个顶点和k条边的标签图的数量，$R(n)$是有性质P和n个顶点的标签图的数量. 假设

$$G_n(x)=\sum_{k=0}^{\infty} c_kx^k$$

295

是普通生成函数，而且我们知道$G_n(x)=(1+x+x^2)^n$. 确定$R(n)$.

12. 假设$d_k=S(n,k)$是有某个性质Q且有n个顶点和k条弧的标签有向图的数量，且$S(n)$是有性质Q和n个顶点且至少两条弧的标签有向图的数量. 设普通生成函数由下式给出:

$$H_n(x)=\sum_{k=0}^{\infty} d_kx^k.$$

假设我们知道$H_n(x)=(1+x^2)^{n+5}$. 确定$S(n)$.

13. 计算图5.4中每一个棋盘的车多项式.

14. 计算$n\times n$棋盘且所有方格都被涂成深色的车多项式，如果n是:

(a)3　　(b)4　　(c)6　　(d)8

15. 拉丁矩形(latin rectangle)是项为1，2，…，n的一个$r\times s$数组，使得任意一行上或列上的项都不相同.

拉丁方(例 1.1)是 $r=s=n$ 的拉丁矩形．构建拉丁方的一个方法是一次建立它的一行，持续地把各行加到拉丁矩形中．我们有多少种方法把图 5.5 中的拉丁矩形加上第三行？通过观察第 j 列中可以包含的符号，作为车多项式问题建立这一拉丁矩形．你不必求解这一问题．

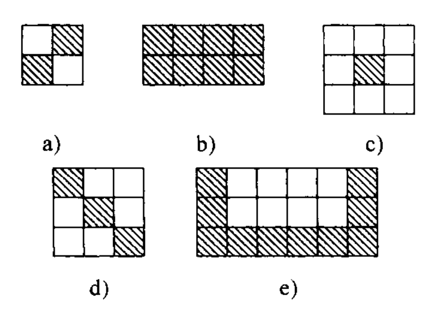


图 5.4　5.1 节练习 13 的棋盘

1	2	3	4	5
2	3	1	5	4

图 5.5　一个 2×5 拉丁矩形

16. 使用车多项式计数 1，2，3，4 的排列的数量，其中 1 不在第二个位置，2 不在第四个位置，3 不在第一个位置和第四个位置．

296

17. 证明：如果棋盘 B' 是通过删除棋盘 B 中没有深色格的行或列而得到的．那么 $r_k(B)=r_k(B')$．

5.2　生成函数的运算

一个序列定义唯一一个生成函数，而一个生成函数又定义唯一一个序列；我们可以在序列和生成函数之间来回转换．编译一个基础生成函数和它们对应的序列的"库"将会十分有用．我们的列表可以开始于生成函数 $1/(1-x)$，e^x，$\sin x$，$\ln(1+x)$，它们所对应的序列可以由等式(5.2)～(5.5)得到．如 5.1.1 节那样通过对这些生成函数进行运算，即通过加、乘、除、合成、微分或积分，可以增加我们的基础列表．这就是本节要做的事情．

本节观察生成函数的各种运算与对应的序列的运算之间的关系．我们先从某些简单的例子开始．假设(a_k)是有普通生成函数 $A(x)=\sum_{k=0}^{\infty} a_k x^k$ 的序列．那么 $A(x)$乘以 x 对应于把这一序列向右移动一个位置并从 0 开始．因为 $xA(x)=\sum_{k=0}^{\infty} a_k x^{k+1}$ 是序列$(0, a_0, a_1, a_2, \cdots)$的普通生成函数．类似地，$A(x)$乘以 $1/x$ 并减去 a_0/x 对应于把这一序列向左移动一个位置以及消去它的第一项，因为

$$\frac{1}{x}A(x)-\frac{a_0}{x}=\sum_{k=0}^{\infty} a_k x^{k-1}-\frac{a_0}{x}=\sum_{k=1}^{\infty} a_k x^{k-1}=\sum_{k=0}^{\infty} a_{k+1} x^k.$$

为了说明这两个结果，注意，因为 $A(x)=e^x$ 是下面序列的普通生成函数：

$$\left(1,1,\frac{1}{2!},\frac{1}{3!},\cdots\right),$$

xe^x 是下面序列的普通生成函数：

$$\left(0,1,1,\frac{1}{2!},\frac{1}{3!},\cdots\right),$$

而$(1/x)e^x-1/x$ 是下面序列的普通生成函数：

$$\left(1,\frac{1}{2!},\frac{1}{3!},\cdots\right).$$

类似地，由等式(5.6)，对 x 求 $A(x)$的导数对应于把(a_k)的第 k 项乘以 k 并向左移动一个位

置．因此，我们已在5.1.1节看到，因为

297

$$\frac{1}{(1-x)^2}=\frac{\mathrm{d}}{\mathrm{d}x}\left[\frac{1}{1-x}\right],$$

所以$1/(1-x)^2$是序列(1，2，3，…)的普通生成函数．

假设(a_k)和(b_k)分别是有普通生成函数$A(x)=\sum_{k=0}^{\infty}a_kx^k$和$B(x)=\sum_{k=0}^{\infty}b_kx^k$的序列．因为两个幂级数可以逐项相加，所以我们看到$C(x)=A(x)+B(x)$是第$k$项为$c_k=a_k+b_k$的序列$(c_k)$的普通生成函数．序列$(c_k)$称为$(a_k)$与$(b_k)$的和，记为$(a_k)+(b_k)$．因此，

$$\frac{1}{1-x}+\mathrm{e}^x$$

是下面序列的普通生成函数：

$$(1,1,1,\cdots)+\left(1,1,\frac{1}{2!},\frac{1}{3!},\cdots\right)=\left(2,2,1+\frac{1}{2!},1+\frac{1}{3!},\cdots\right).$$

从组合数学的观点看，两个生成函数相乘会发生最有趣的情况．假设

$$C(x)=A(x)B(x),\tag{5.11}$$

其中$A(x)$，$B(x)$和$C(x)$分别是序列(a_k)，(b_k)和(c_k)的普通生成函数．对于所有的k都有$c_k=a_kb_k$成立吗？设$A(x)=1+x$，$B(x)=1+x$，那么$C(x)=A(x)B(x)=1+2x+x^2$．现在$(c_k)=(1,2,1,0,0,\cdots)$且$(a_k)=(b_k)=(1,1,0,0,\cdots)$，所以$c_0=a_0b_0$但是$c_1\neq a_1b_1$．因此，从(5.11)式不能得到$c_k=a_kb_k$．通过观察可以发现如果用$A(x)$去乘以$B(x)$，通过把$A(x)$的项$a_jx^j$和$B(x)$的项$b_{k-j}x^{k-j}$结合，可以得到项$c_kx^k$．因此，对于所有的$k$，

$$c_k=a_0b_k+a_1b_{k-1}+\cdots+a_{k-1}b_1+a_kb_0.\tag{5.12}$$

在$A(x)$和$B(x)$都等于$1+x$时，很容易核实这一情况．注意，如果$k=0$，(5.12)式说明$c_0=a_0b_0$．同样注意对于所有的k，(5.12)式蕴涵(5.11)式．如果(5.12)式对于所有的k都成立，我们说序列(c_k)是两个序列(a_k)和(b_k)的*卷积*(convolution)，并写作$(c_k)=(a_k)*(b_k)$．我们的结果概括如下．

定理5.1 *假设$A(x)$，$B(x)$和$C(x)$分别是序列(a_k)，(b_k)和(c_k)的普通生成函数．那么有*

(a)$C(x)=A(x)+B(x)$当且仅当$(c_k)=(a_k)+(b_k)$．

(b)$C(x)=A(x)B(x)$当且仅当$(c_k)=(a_k)*(b_k)$．

【例5.11】 假设对于所有的k有$b_k=1$．那么(5.12)式变成

$$c_k=a_0+a_1+\cdots+a_k.$$

生成函数$B(x)$由$B(x)=1/(1-x)=(1-x)^{-1}$给出．因此，根据定理5.1，有

298

$$C(x)=A(x)(1-x)^{-1}.$$

这是一个级数的前k项和的生成函数．例如，假设(a_k)是序列(0，1，1，0，0，…)．那么$A(x)=x+x^2$且

$$\begin{aligned}C(x)&=(x+x^2)[1+x+x^2+\cdots]\\&=x+2x^2+2x^3+2x^4+\cdots.\end{aligned}$$

我们得出结论：

$$(x+x^2)(1-x)^{-1}$$

是序列$(c_k)=(0,1,2,2,\cdots)$的普通生成函数．这可以通过注意下面的事实而得到证实：

$$a_0=0,a_0+a_1=1,a_0+a_1+a_2=2,a_0+a_1+a_2+a_3=2,\cdots$$ ■

【例5.12】 如果$A(x)$是序列(a_k)的生成函数，那么$A^2(x)$是序列(c_k)的生成函数，其中

$$c_k = a_0a_k + a_1a_{k-1} + \cdots + a_{k-1}a_1 + a_ka_0.$$

在 6.4 节中，这一结果也将用于通过计数树枚举化学异构. 特别地，如果对于所有的 k，有 $a_k=1$，那么 $A(x)=(1-x)^{-1}$. 这时，

$$C(x) = A^2(x) = (1-x)^{-2}$$

是序列(c_k)的生成函数，其中 $c_k=k+1$. 之前我们已经通过微分$(1-x)^{-1}$得到这一结果. ■

【例 5.13】 假设

$$G(x) = \frac{1+x+x^2+x^3}{1-x}$$

是序列(a_k)的普通生成函数. 我们能够得到 a_k 吗? 我们可以把 $G(x)$写成如下形式:

$$G(x) = (1+x+x^2+x^3)(1-x)^{-1}.$$

现在 $1+x+x^2+x^3$ 是下面序列的普通生成函数:

$$(b_k) = (1,1,1,1,0,0,\cdots),$$

而且$(1-x)^{-1}$是下面序列的普通生成函数:

$$(c_k) = (1,1,1,\cdots).$$

因此，$G(x)$是这两个序列的卷积的普通生成函数，即

$$a_k = b_0c_k + b_1c_{k-1} + \cdots + b_kc_0 = b_0 + b_1 + \cdots + b_k.$$

根据这一表达式很容易证明

$$(a_k) = (1,2,3,4,4,\cdots).$$ ■ 299

【例 5.14 车多项式的化简】 在计算车多项式时，把一个复杂的计算化简成若干更小的计算常常很有用，这是我们在 3.4 节有关色多项式的内容中遇到过的一个技巧. 5.1 节的练习 17 证明了这样的一个化简. 这里我们给出另一个化简. 假设 I 是棋盘 B 中涂成深色的方格的一个集合，而 B_I 是通过把不在 I 中的深色方格的颜色变淡从 B 得到的棋盘. 假设 B 中的深色方格被划分成两个集合 I 和 J，使得 I 中的所有方格都不与 J 的任意方格位于同一行或同一列. 对于这种情况，我们说 B_I 和 B_J 分解(decompose)B. 图 5.6 说明这一情况.

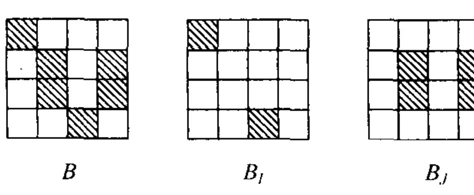


图 5.6　B_I 和 B_J 分解 B

如果 B_I 和 B_J 分解 B，那么因为所有可接受方格都落入到 B_I 或 B_J 中，且 I 中的车不可以吃掉 J 中的车，反之也一样，为了在 B 上放置 k 个吃不掉的车，对于某个 p，我们在 B_I 上放置 p 个吃不掉的车，然后在 B_J 上放置 $k-p$ 个吃不掉的车. 因此，

$$r_k(B) = r_0(B_I)r_k(B_J) + r_1(B_I)r_{k-1}(B_J) + \cdots + r_p(B_I)r_{k-p}(B_J) + \cdots + r_k(B_I)r_0(B_J).$$

这表明序列$(r_k(B))$的确是序列$(r_k(B_I))$和$((r_k(B_J))$的卷积. 因此

$$R(x,B) = R(x,B_I)R(x,B_J).$$ ■

本节练习

1. 下列每一个函数都是某个序列(a_k)的普通生成函数. 确定这一序列.

(a)$x\ln(1+x)$　　(b)$\frac{1}{x}\sin x$　　(c)$x^4\ln(1+x)$

(d)$\frac{1}{x^4}\sin x$　　(e)$\frac{5}{1-x}+x^3+3x+4$　　(f)$\frac{x}{1-7x}+\frac{4}{1-x}$

(g) $\frac{1}{1-x^2}+6x+5$　　(h) $\frac{1}{2}(e^x-e^{-x})$

[300] 2. 对于下列每一个函数 $A(x)$，假设 $B(x)=xA'(x)$. 寻找 $B(x)$是普通生成函数的序列.

(a) $\frac{1}{1-x}$　　(b) e^{3x}　　(c) $\cos x$　　(d) $\ln(1+x)$

3. 求下列各组序列的卷积的公式：

(a)(1, 1, 1, …)和(1, 1, 1, …)

(b)(1, 1, 1, …)和(0, 1, 2, 3, …)

(c)(1, 1, 0, 0, …)和(0, 1, 2, 3, …)

(d)(1, 2, 4, 0, 0, …)和(1, 2, 3, 4, 0, …)

(e)(1, 0, 1, 0, 0, 0, …)和(2, 4, 6, 8, …)

(f)(0, 0, 0, 1, 0, 0, …)和(8, 9, 10, 11, …)

4. 下列每一个函数都是某个序列(a_k)的普通生成函数. 确定这一序列.

(a) $\left(\frac{5}{1-x}\right)\left(\frac{3}{1-x}\right)$　　(b) $\frac{1}{1-x}\ln(1+2x)$　　(c) $\frac{x^3+x^5}{1-x}$

(d) $\frac{x^2-3x}{1-x}+x$　　(e) $(1+x)^q$，其中 q 是正整数　　(f) $xe^{3x}+(1+x)^2$

5. 如果 $G(x)=[1/(1-x)]^2$ 是序列(a_k)的普通生成函数，确定 a_4.

6. 如果 $A(x)=(1-5x^2)(1+2x+3x^2+4x^3+\cdots)$是序列$(a_k)$的普通生成函数，确定 a_{11}.

7. 假设 $A(x)$是序列(1, 3, 9, 27, 81, …)的普通生成函数，且 $B(x)$是序列(b_k)的普通生成函数. 如果 $B(x)$是下列函数，确定(b_k)：

(a) $A(x)+x$　　(b) $A(x)+\frac{1}{1-x}$　　(c) $2A(x)$

8. 在下列每一种情况中，假设 $B(x)$是(b_k)的普通生成函数，且 $A(x)$是(a_k)的普通生成函数. 通过$A(x)$确定 $B(x)$的表达式.

(a) $b_k=\begin{cases} a_k & \text{如果 } k\neq 3 \\ 11 & \text{如果 } k=3 \end{cases}$　　(b) $b_k=\begin{cases} a_k & \text{如果 } k\neq 0,\ 4 \\ 2 & \text{如果 } k=0 \\ 1 & \text{如果 } k=4 \end{cases}$

9. 假设

$$a_k=\begin{cases} b_k & \text{如果 } k\neq 0,2 \\ 4 & \text{如果 } k=0 \\ 1 & \text{如果 } k=2. \end{cases}$$

$B(x)$是序列(b_k)的普通生成函数，如果 $b_0=2$，$b_2=0$. 通过 $B(x)$确定序列(a_k)的普通生成函数 $A(x)$的表达式.

[301] 10. 求下列各序列(a_k)的普通生成函数的简单、闭式的表达式.

(a) $a_k=k+3$　　(b) $a_k=8k$　　(c) $a_k=3k+4$

11. 利用导数确定下列序列(b_k)的普通生成函数.

(a) $b_k=k^2$　　(b) $b_k=k(k+1)$　　(c) $b_k=(k+1)\frac{1}{k!}$

12. 假设 $A(x)$是序列(a_k)的普通生成函数且 $b_k=a_{k+1}$. 确定序列(b_k)的普通生成函数.

13. 假设

$$a_k=\begin{cases} \sum_{i=0}^{k-2} b_i b_{k-2-i} & \text{如果 } k\geqslant 2 \\ 0 & \text{如果 } k=0 \text{ 或 } k=1. \end{cases}$$

假设 $A(x)$是序列(a_k)的普通生成函数，而 $B(x)$是序列是(b_k)的普通生成函数. 利用 $B(x)$确定 $A(x)$的表达式.

14. 假设 $A(x)$是序列(a_k)的普通生成函数，而且序列(b_k)的定义如下：

$$b_k = \begin{cases} 0 & \text{如果 } k < i \\ a_{k-i} & \text{如果 } k \geq i. \end{cases}$$

利用 $A(x)$确定序列(b_k)的普通生成函数.

15. 确定第 k 项是

$$a_k = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{k!}$$

的序列的普通生成函数 .

16. (a)使用例(5.14)的化简结果计算图 5.4 中棋盘(d)的车多项式.

(b)扩展到对角线上都是深色方格的 $n\times n$ 棋盘.

17. 使用 5.1 节练习 17 的结果，确定图 5.6 中的棋盘 B_I 和 B_J 的车多项式. 然后使用例 5.14 的化简结果计算图 5.6 中的棋盘 B 的车多项式.

5.3 计数的应用

5.3.1 取样问题

生成函数有助于我们解决计数问题. 为了说明生成函数如何帮助计数，我们首先考虑下面的取样问题：取样的对象有不同的类型，而且相同类型的对象是不可区分的. 用 2.9 节的术语，我们考虑的是无放回取样. 例如，假设存在三个对象 a，b 和 c，且每一个对象都可以被选取也可以不被选取. 有多少种可能的选择方法呢？令 a_k 是选择 k 个对象的方法数量. 令 $G(x)$是生成函数$\sum a_k x^k$. 现在很容易看到 $a_k=\binom{3}{k}$，因此，

302

$$G(x) = \binom{3}{0}x^0 + \binom{3}{1}x^1 + \binom{3}{2}x^2 + \binom{3}{3}x^3. \tag{5.13}$$

让我们考虑另一种计算 $G(x)$的方法. 我们可以或者不选择 a 或者选择一个 a，或者不选择 b 或者选择一个 b，或者不选择 c 或者选择一个 c. 让我们考虑下面的乘积：

$$[(ax)^0 + (ax)^1][(bx)^0 + (bx)^1][(cx)^0 + (cx)^1], \tag{5.14}$$

其中加和乘分别对应于“或”和“与”(回想加法规则和乘法规则). 表达式(5.14)变成

$$(1 + ax)(1 + bx)(1 + cx),$$

这一表达式等于

$$1 + (a + b + c)x + (ab + ac + bc)x^2 + abcx^3. \tag{5.15}$$

注意，x 的系数给出得到一个对象的方法：它是 a 或 b 或 c. x^2 的系数给出得到两个对象的方法：它是 a 和 b，或 a 和 c，或 b 和 c，而 x^3 的系数给出得到三个对象的方法，而 x^0 的系数(即 1)给出得不到对象的方法. 如果我们令 $a=b=c=1$，那么 x^k 的系数将计数得到 k 个对象的方法数量，即 a_k. 因此，在(5.15)式中令 $a=b=c=1$ 得到下面的生成函数：

$$G(x) = 1 + 3x + 3x^2 + x^3,$$

这一表达式就是我们在(5.13)式中计算得到的.

同样的技术也适用于 $G(x)$的系数不明确的情况. 这时，我们可以通过这一技术计算$G(x)$，并通过 $G(x)$计算适当的系数.

【例 5.15】 假设我们有三种类型(类型 a、类型 b 和类型 c)的对象. 假设我们可以从类型 a 选择 0、1 或 2 个对象，然后从类型 b 选择 0 或者 1 个对象，最后从类型 c 选择 0 或者 1 个对象. 有多少种选择 k 个对象的方法呢？其答案不是$\binom{4}{k}$. 例如，两个 a 和 1 个 b 不同于 1 个 a，1 个 b

和 1 个 c. 然而，选择第一个 a 和一个 b 等同于选择第二个 a 和一个 b：类型 a 的对象是不可区分的. 我们想要知道取 k 个对象的可区分方法的数量. 假设 b_k 是所要的方法数量. 我们尝试计算普通生成函数 $G(x)=\sum b_k x^k$. 这里要考虑的正确表达式是

$$[(ax)^0+(ax)^1+(ax)^2][(bx)^0+(bx)^1][(cx)^0+(cx)^1], \tag{5.16}$$

因为我们可以选择 0，1 或 2 个类型 a 的对象，0 或 1 个类型 b 的对象，0 或 1 个类型 c 的对象. 表达式(5.16)可以化简成

$$(1+ax+a^2x^2)(1+bx)(1+cx),$$

这一表达式等于

303

$$1+(a+b+c)x+(ab+bc+ac+a^2)x^2+(abc+a^2b+a^2c)x^3+a^2bcx^4. \tag{5.17}$$

正如例 5.14 中那样，x^3 的系数给出得到三个对象的方法：一个 a、一个 b 和一个 c，或者 2 个 a 和一个 b，或者 2 个 a 和一个 c. 对于其他的系数这一事实也成立. 另外，在(5.17)式中取 $a=b=c=1$ 给出下面的生成函数：

$$G(x)=1+3x+4x^2+3x^3+x^4.$$

x^k 的系数是 b_k. 例如，$b_2=4$(读者应该核实为什么是这样的). ■

一般地，假设我们有 p 种类型的对象，且有 n_1 个类型 1 的对象，n_2 个类型 2 的对象，…，n_p 个类型 p 的对象. 令 c_k 是选择 k 个对象的不同方法的数量，如果我们可以选择每一种类型的任意多个对象. 其普通生成函数由 $G(x)=\sum c_k x^k$ 给出. 为了计算这一生成函数，我们考虑下面的乘积：

$$[(a_1x)^0+(a_1x)^1+\cdots+(a_1x)^{n_1}]\times[(a_2x)^0+(a_2x)^1+\cdots+(a_2x)^{n_2}]\times\cdots$$
$$\times[(a_px)^0+(a_px)^1+\cdots+(a_px)^{n_p}].$$

令 $a_1=a_2=\cdots=a_p=1$，我们得到

$$G(x)=(1+x+x^2+\cdots+x^{n_1})(1+x+x^2+\cdots+x^{n_2})\cdots(1+x+x^2+\cdots+x^{n_p}).$$

数 c_k 由 $G(x)$ 中 x^k 的系数给出. 因此，我们有下面的定理.

定理 5.2　*假设我们有 p 种类型的对象，且每一类型 i 有 n_i 个不可区分的对象，$i=1$，2，…，p. 如果我们可以选择每一种类型的任意多个对象，那么选择 k 个对象的不同方法的数量由下面普通生函数 $G(x)$ 中的 x^k 的系数给出：*

$$G(x)=(1+x+x^2+\cdots+x^{n_1})(1+x+x^2+\cdots+x^{n_2})\cdots(1+x+x^2+\cdots+x^{n_p}).$$

【例 5.16 不可区分的男士和女士】　假设我们有 m 位(不可区分)男士和 n 位(不可区分)女士. 如果我们能够选取任意数量的男士和任意数量的女士，那么定理 5.2 表明我们可以选 k 个人的方法数量由下面 $G(x)$ 中的 x^k 的系数给出：

$$G(x)=(1+x+\cdots+x^m)(1+x+\cdots+x^n). \tag{5.18}$$

注意　*这一系数不是*

$$\binom{m+n}{k},$$

这是因为，例如，不论你如何选择男士和女士，取 3 位男士和 $k-3$ 位女士的方法是一样的. 现在，$1+x+\cdots+x^m$ 是下面序列的生成函数：

304

$$(a_k)=(1,1,\cdots,1,0,0,\cdots),$$

且 $1+x+\cdots+x^n$ 是下面序列的生成函数：

$$(b_k)=(1,1,\cdots,1,0,0,\cdots),$$

其中对于 $k=0$，1，…，m，$a_k=1$，对于 $k=0$，1，…n，$b_k=1$. 根据(5.18)式可知，

$G(x)$是序列(a_k)和(b_k)的卷积的生成函数. 我们把这一卷积的一般计算留给出读者(练习 11). 例如, $m=8$, $n=7$ 时, 我们选择 $k=9$ 个人的方法数由下式给出:

$$a_0b_9+a_1b_8+\cdots+a_9b_0=0+0+1+\cdots+1+1+0=7.$$

作为对这一答案的验证, 我们注意到这七种方法如下: 2 位男士和 7 位女士, 3 位男士和 6 位女士, …, 8 位男士和 1 位女士. ■

【例 5.17 抽样调查】 在做抽样调查时, 假设我们已把采访的可能男士分成各种范畴, 例如, 教师、医生、律师等等, 对女士也类似. 假设在我们的分组中, 我们在每一范畴中选取两位男士和一位女士, 且假设有 q 种范畴. 存在多少种不同的方法取 k 个人的样本呢? 现在, 我们想要区分相同性别的人, 当且仅当他们属于不同的范畴. 选择 k 个人的方法数量的生成函数由下式给出:

$$G(x)=\underbrace{(1+x+x^2)(1+x+x^2)\cdots(1+x+x^2)}_{q\text{项}}\underbrace{(1+x)(1+x)\cdots(1+x)}_{q\text{项}}$$
$$=(1+x+x^2)^q(1+x)^q.$$

$G(x)$的各项来自于下面的事实: 在 q 种范畴的每一个中, 我们或者取 0, 1, 或 2 位男士作样本, 因此对每一个范畴给出一个$(1+x+x^2)$项; 另外, 我们选择 0 或 1 位女士, 所以对 q 种范畴中的每一个范畴我们有一个$(1+x)$项. 选择 k 个人的方法数量是$G(x)$中 x^k 的系数. 例如, 如果 $q=2$, 那么

$$G(x)=x^6+4x^5+8x^4+10x^3+8x^2+4x+1.$$

例如, 选择 3 个人有 10 种方法. 读者也许希望确认这 10 种方法. 一般地, 如果存在 m 种范畴的男士及 n 种范畴的女士, 且范畴 i 中有 p_i 位男士, $i=1$, 2, $\cdots m$, 范畴 j 中有 q_j 位女士, $j=1$, 2, …, n, 读者也许希望推演选取 k 个人的方法数量的普通生成函数. ■

【例 5.18 另一种调查】 假设我们调查小组正在考虑三个家庭. 第一个家庭有两个人(称他们为(家庭)a 的人). 第二家庭有一个人(称那个人为 b). 第三个家庭有一个人(称那个人为 c). 如果对于一个给定的家庭, 我们或者一个人都不选取或者选取那个家庭的所有成员, 那么选择 k 个人有多少种方法呢? 让我们再一次考虑乘法规则和加法规则: 我们可以选择 0 个或 2 个 a, 选取 0 个或者 1 个 b, 选取 0 个或者 1 个 c. 因此, 我们考虑下面的表达式:

305

$$[(ax)^0+(ax)^2][(bx)^0+(bx)^1][(cx)^0+(cx)^1].$$

其结果是

$$(1+a^2x^2)(1+bx)(1+cx)=1+(b+c)x+(bc+a^2)x^2+(a^2b+a^2c)x^3+a^2bcx^4.$$

选取两个人的方法由 x^2 的系数给出: 我们可以选取 b 和 c, 或者选取两个 a. 设 $a=b=c=1$, 我们可以得到生成函数

$$G(x)=1+2x+2x^2+2x^3+x^4.$$

选取 k 个人的方法数量由 x^k 的系数给出. ■

【例 5.19】 假设我们有 p 种不同的对象, (实际上)每一种类型的对象的供给不受限制. 有多少种方法选取 k 个对象的样本呢? 答案由下面的普通生成函数中的 x^k 的系数给出:

$$G(x)=\underbrace{(1+x+x^2+\cdots)(1+x+x^2+\cdots)\cdots(1+x+x^2+\cdots)}_{p\text{项}}$$
$$=(1+x+x^2+\cdots)^p \tag{5.19}$$
$$=(1-x)^{-p}\qquad[\text{由}(5.2)\text{式}]$$

(最终给出相同结果的另一个方法是对于 $n_i=k$ 运用定理 5.12). 我们想要开发已知诸如(5.19)式

这样的表达式 $G(x)$ 确定 x^k 的系数的方法. 为此，我们在下一节介绍二项式定理. ■

【例 5.20 方程的整数解】 假设对于 $i=1, 2, 3$，$b_i \geq 0$. 下面的方程存在多少个整数解?

$$b_1 + b_2 + b_3 = 14,$$

因为每一个 b_i 可以取值 0 或 1 或 2，…，由例 5.19，这一问题的答案由下面的普通生成函数中 x^{14} 的系数给出.

$$\begin{aligned}G(x) &= (1+x+x^2+\cdots)(1+x+x^2+\cdots)(1+x+x^2+\cdots)\\ &= (1+x+x^2+\cdots)^3\\ &= (1-x)^{-3},\end{aligned}$$

306 ■

【例 5.21 编码理论(再探例 2.27)】 在例 2.27 中，我们考虑了在传输后核实代码字的问题. 这里，我们考虑代码字作为位串的一般情况. 假设我们检查三种类型的错误：增加一个数字(0 或 1)、减少一个数字(0 或 1)和一个数字的逆转(0 到 1 或 1 到 0). 有多少种方式可以找到 30 个错误呢? 为了回答这一问题，我们假设相同类型的两个错误是不可区分的. 另外实际上不限制每一种类型错误的出现次数. 因此，我们寻找下面普通生成函数中 x^{30} 的系数：

$$G(x) = (1+x+x^2+\cdots)(1+x+x^2+\cdots)(1+x+x^2+\cdots) = (1-x)^{-3}.$$

这是(5.19)式的特殊情况.

接下来假设我们不区分三种类型的错误，但是我们要记录在输送到的第一个代码字、第二个代码字等中是否出现错误. 如果有 100 个代码字，那么有多少种不同的方式找到 30 个错误? 这就是我们在例 2.27 中所陈述的问题. 为了回答这一问题，我们注意有 100 种类型的错误，每一个代码字对应一种类型. 我们可以或者假设每一个代码字可能的错误数量实际上是无限的，或者它以 30 为界，或者它以每一个代码字中的数字数量为界. 对于前面的情况，我们考虑下面的普通生成函数：

$$G(x) = (1+x+x^2+\cdots)^{100} = (1-x)^{-100},$$

并寻找 x^{30} 的系数. 对后面两种情况，我们只是让 $G(x)$ 中的各项结束于 x 的适当幂，当然，如果每个代码字至少有 30 位，那么无论代码字具体是多少位，x^{30} 的系数都是不变的. ■

【例 5.22 Sicherman 骰子】 假设要投掷一对标准的骰子. 该投掷的各种结果的概率是多少? 因为每一个骰子可以取的值为 1，2，3，4，5，6，所以我们考虑下面的普通生成函数：

$$\begin{aligned}p(x) &= (x+x^2+x^3+x^4+x^5+x^6)(x+x^2+x^3+x^4+x^5+x^6)\\ &= (x^2+2x^3+3x^4+4x^5+5x^6+6x^7+5x^8+4x^9+3x^{10}+2x^{11}+x^{12}).\end{aligned}$$

注意，x^j 的系数是得到 j 的一次投掷的方法数量. 同样系数和 36 给出了两个骰子的不同投掷的可能的总数量. 所以投掷出一个 5 的概率是 4/36.

其他的骰子的情况又如何呢? 存在一对六侧面不同的骰子，它产生与标准骰子相同结果概率吗? 我们只考虑带有正整数标签的骰子. 设 a_1，a_2，a_3，a_4，a_5，a_6 是其中一个骰子上的值，而 b_1，b_2，b_3，b_4，b_5，b_6 是另外一个骰子上的值. 如果另一对骰子存在，那么它必须是下面的情况：

$$\begin{aligned}&(x+x^2+x^3+x^4+x^5+x^6)(x+x^2+x^3+x^4+x^5+x^6)\\ &= (x^{a_1}+x^{a_2}+x^{a_3}+x^{a_4}+x^{a_5}+x^{a_6})(x^{b_1}+x^{b_2}+x^{b_3}+x^{b_4}+x^{b_5}+x^{b_6}).\qquad(5.20)\end{aligned}$$

307

(5.20)式的左边 $p(x)$ 因式分解成

$$x^2(1+x)^2(1+x+x^2)^2(1-x+x^2)^2.$$

因为有整数系数的多项式的因式分解总是唯一的(这是任何抽象代数课程都要讲的标准事实)，

所以(5.20)式的右边也必须有这样的因式分解. 因此,

$$f(x)=(x^{a_1}+x^{a_2}+x^{a_3}+x^{a_4}+x^{a_5}+x^{a_6})=x^{c_1}(1+x)^{c_2}(1+x+x^2)^{c_3}(1-x+x^2)^{c_4},$$

而且

$$g(x)=(x^{b_1}+x^{b_2}+x^{b_3}+x^{b_4}+x^{b_5}+x^{b_6})=x^{d_1}(1+x)^{d_2}(1+x+x^2)^{d_3}(1-x+x^2)^{d_4},$$

其中, 对于 $i=1, 2, 3, 4$, $c_i+d_i=2$.

为了寻找 c_i 和 d_i(因此 a_i 和 b_i), 首先考虑

$$f(0)=(0^{a_1}+0^{a_2}+0^{a_3}+0^{a_4}+0^{a_5}+0^{a_6}).$$

这一等式等于 0, $g(0)$也等于 0. 所以 x 项必须同时出现在 $f(x)$和 $g(x)$的因式分解式中. 因此, c_1 和 d_1 必须同时等于 1. 接下来, 考虑 $f(1)=(1^{a_1}+1^{a_2}+1^{a_3}+1^{a_4}+1^{a_5}+1^{a_6})=6$ 或 $1^{c_1}2^{c_2}3^{c_3}1^{c_4}$. 所以 $c_2=c_3=1$. 由类似的分析和结论可知: 对于 $g(x)$, $d_1=d_2=1$ 也成立. 最后, 我们考虑 c_4 和 d_4. 或者 $c_4=d_4=1$, 这产生一对标准的骰子, 或者 $c_4=0$, $d_4=2$(反之亦然). 在后面的情况中, 多项式相乘之后, 我们看到

$$f(x)=x+x^2+x^2+x^3+x^3+x^4,\quad g(x)=x+x^3+x^4+x^5+x^6+x^8$$

及

$$f(x)g(x)=(x^2+2x^3+3x^4+4x^5+5x^6+6x^7+5x^8+4x^9+3x^{10}+2x^{11}+x^{12}).$$

这一对骰子中, 一个标记为 1, 2, 2, 3, 3, 4, 另一个标记为 1, 3, 4, 5, 6, 8, 称为 Sicherman 骰子⊖. Sicherman 骰子是与标准骰子的结果概率相同的唯一的另一对(带有正整数标签的)骰子. ■

【例 5.23 整数划分】 回顾 2.10.5 节, 一个正整数 k 的划分是和等于 k 的正整数集合. 例如, 整数 4 有划分{1, 1, 1, 1}, {1, 1, 2}, {2, 2}, {1, 3}和{4}. 假设 $p(k)$是整数 k 的划分的数量. 因此, $p(4)=5$. 练习 12～16 利用本节的技术研究整数的划分. 这一工作的想法要追溯到 1748 年的欧拉. 关于划分的详细讨论, 可以参见大多数数论书籍: 例如 Bressoud 和 Wagon [2000]、Hardy 和 Wright[1980]. 也可参见 Berge[1971]、Cohen[1978]、Tomescu[1985]. ■ 308

【例 5.24 构成计算机二进制运算基础的关键观察: 划分成不同的整数】 设 $p^*(k)$是把整数 k 划分成不同整数的方法数量. 因此, 使用划分{7}, {1, 6}, {2, 5}, {3, 4}, {1, 2, 4}, $p^*(7)=5$. 成为构成遍及计算机的二进制运算基础的一个关键性的观察是: 每一个整数都可以划分成为 2 的不同幂的和. 例如, $7=2^0+2^1+2^2$, $19=2^0+2^1+2^4$. 在练习 16(c)中, 我们要求读者利用本节的方法证明这一点. ■

5.3.2 关于分装问题的一个注释

在 2.10 节, 我们考虑了分装问题, 这是一类把球分配到盒子里的问题. 例 5.21 的第二部分涉及分装问题: 我们有 30 个球(错误)和 100 个盒子(代码字). 这是下面分装问题的特殊情况: 其中我们有 k 个不可区分的球, 且我们希望把它们分配到 p 个可区分的盒子里. 如果我们不对每一个盒子里的球数加以限制, 那么很容易扩展例 5.21 中的推理, 并证明把 k 个球分配到 p 盒子的方法数量由(5.19)式的普通生成函数中 x^k 的系数给出. 相反, 如果我们不允许在第 i 盒子里的球的数量超过 n_i, 那么我们容易看到, 定理 5.2 中的普通生成函数中 x^k 的系数给出这一方法的数量.

⊖ George Sicherman 首先考虑并解决了其结果概率与标准骰子的概率相同的其他可能的骰子问题. 对这一问题的讨论, 参见 Gardner[1978].

本节练习

1. 对于下列每一种情况，建立适当的生成函数. 不计算结果，但要指出你在寻找的内容：例如 x^{10} 的系数.

   (a)一个调查小组想要从密歇根大学选出至多三名男生，从布朗大学至多选出 3 名女生，从斯坦福大学至多选出 2 名男生，从赖斯大学至多选出 2 名女生. 如果我们只能从密歇根大学和斯坦福大学选出男生，从布朗大学和赖斯大学选出女生，而且从同一学校选出的两名同性的学生是不可区分的，那么我们有多种方法选出 5 名学生调查？

   (b)如果从 a，b，c，d 中选出 5 个字母，而且 b，c 和 d 至多只能选择一次，如果选取 a 的话必须选取 4 次，那么有多少种选择方法？

   (c)在策划一次考试时，一位主考老师想要出至少 3 道容易题、至少 3 道中等难度的题和至少 2 道难题. 她的选择限制(上限)是 7 道容易题、6 道中等难度的题和 4 道难题. 她有多少种方法选出 11 道题？(考题顺序在选定题后决定.)(注意：两道同样难度的题目之间是不可区分的.)

   309 (d)如果要选出 8 个二进制数字，而且每一个数字都必须出现偶数次，那么有多少种选择方法？

   (e)从 6 名共和党人、6 名民主党人和 7 名无党派人士中选出 12 名投票人，如果至少包括 4 名无党派人士且相同党派的任意两名投票人不可区分，那么有多少种选择方法？

   (f)Geiger 计数器记录五种不同种类的放射性粒子在 5 分钟内的碰撞. 有多少种方法可以得到 20 个记录？

   (g)在检查一台通信设备的工作状况时，我们要找出 4 种类型的传输错误. 我们有多少方法可以找到 40 个错误？

   (h)在(g)中，假设我们不区分错误类型，但是，我们记录错误出现的日期. 我们有多少种不同方法可以在 100 天内找到 40 个错误？

   (i)有多少种方法可以把 12 个不可区分的球分配到 8 个可区分的盒子里？

   (j)如果每一个盒子都不能是空的，重复(i).

   (k)如果投掷 14 个标准骰子，有多少种方法使总点数等于 30？

   (l)一个调查小组根据年龄可把被调查者分成 6 组，且彼此独立地，根据地理区域可把这些人分成 5 组. 如果 2 个人是可区分的仅当他们属于不同的年龄组，生活在不同的地理区域，或性别相反，那么有多少种方法选出 10 名被调查者？

   (m)把一美元兑换成硬币(一分，五分，一角，二角五分)的方法数是多少？

   (n)确定下面方程的解的个数：

   $$x_1 + x_2 + x_3 = 14,$$

   其中，各 x_i 是非负整数，且 $x_i \leqslant 7$.

   (o)确定下面方程的解的个数：

   $$x_1 + x_2 + x_3 = 20,$$

   其中，对于所有的 i，有 $x_i \geqslant 0$，且 x_1 是奇数，$2 \leqslant x_2 \leqslant 5$，$x_3$ 是素数.

2. 假设我们想要构建长度为 5，且 U、C 和 A 分别至多使用一次而 G 可以使用任意次的一条 RNA 链. 有多少种方法选取这些基(不考虑顺序)？通过构建一个生成函数并计算一个适当的系数来回答这一问题.

3. 一位顾客想要买六个水果，其中包含至多两个苹果、至多两个桔子、至多两个梨和至少一个、至多两个桃. 如果任意两个相同类型的水果，如两个桃是不可区分的，那么有多少种方法买六个水果？

4. 假设有 p 种对象，且第 i 种有 n_i 个不可区分的对象. 假设对于每一种对象我们可以全选或不选. 构建一个计算选出 k 个对象的方法数量的生成函数.

310 5. 考虑 2.15 节练习 7 的投票情况.

   (a)如果一个省的代表投票都一样，构建一个计算得到 k 张投票的方法数量的生成函数(可以认为从省 A 的代表得到的投票不同于从省 B 的代表得到的投票，以此类推).

   (b)如果一个省的代表不必投相同的票，重复(a).

6. 考虑下面来自于《食物指南金字塔》的基本食物组别：面包、水果、蔬菜、肉、牛奶和脂肪. 一位营养学家想要为一个自助餐厅(出于服务空间的限制)选出 10 项作为每日菜单，他至少在每一个范畴选取一项.
   (a)如果认为相同组别中的项是不可区分的，且假设，为了实用的目的，在每一个组别中食物的数量足够大，那么有多少种方法来选择基本菜单？通过构建生成函数来回答这一问题，不必计算.
   (b)如果认为一个组别中的各项是可区分的，而且每一组别中有 30 项，那么如何处理这一问题？
7. 假设存在 p 种对象，且每一种对象的供给不受限制. 设 a_k 是选择 k 个对象的不同方法，如果每一种对象只能选择偶数个(包括 0 个)，构建 a_k 的生成函数.
8. 在一次包含新英格兰地区各州的总统初选中，一名总统候选人得到的来自各州的选举团投票的数量与各州内投票于这一候选人的投票人的数量成正比. 例如，在缅因州得到 50%的投票而其他州没有得到投票的候选人，按惯例将获得两张选举团投票(或两张州投票)，因为缅因州有四张选举团选票. 投票将通过四舍五入化整，因为没有小数投票. 构建计算一名候选人得到 25 张选举团投票的方法数量的生成函数. (根据 2000 年人口普查，这一地区各州的选举团投票的票数是：康涅狄格 7；缅因州 4；马萨诸塞州 12；新罕布什尔州 4；罗德岛州 4；佛蒙特州 3.)
9. 在例 5.17 中，我们已证明如果 $q=2$，有 10 种方法选出 3 个人. 它们分别是什么？
10. (Gallian[2002])假设你有一个 18 面骰子，其标签分别是 1，2，2，3，3，3，4，4，4，5，5，5，6，6，6，7，7，8.
    (a)寻找一个 2 面骰子上的标签，使得当投掷这一骰子和上面的 18 面骰子时，该骰子具有与一对标准 6 面骰子同样的结果概率.
    (b)扩展(a)，寻找一个 4 面骰子的标签，使得当投掷这一骰子和上面的 18 面骰子时，该骰子具有与一对标准的 6 面骰子同样的结果概率.
11. 求例 5.16 中确定的两个序列(a_k)和(b_k)的卷积的一般公式.
12. 接下来的 5 个练习都是研究例 5.23 和例 5.24 定义的整数划分的. 设 $p(k)$是整数 k 的划分数量，且设

$$G(x)=\frac{1}{(1-x)(1-x^2)(1-x^3)(1-x^4)\cdots}.$$

    证明 $G(x)$是序列$(p(k))$的普通生成函数.

311

13. 回想，$p^*(k)$是把整数 k 划分成不同整数的方法数.
    (a)确定 $p^*(8)$.
    (b)确定 $p^*(11)$.
    (c)确定$(p^*(k))$的普通生成函数.
14. 设 $p_o(k)$是把整数 k 划分成不必互不相同的奇整数的方法数.
    (a)确定 $p_o(7)$.
    (b)确定 $p_o(8)$.
    (c)确定 $p_o(11)$.
    (d)确定$(p_o(k))$的生成函数.
15. 设 $p^*(k)$和 $p_o(k)$分别如练习 13 和练习 14 所定义的那样. 证明 $p^*(k)=p_o(k)$.
16. (a)证明对于 $|x|<1$，有

$$(1-x)(1+x)(1+x^2)(1+x^4)(1+x^8)\cdots(1+x^{2^k})\cdots=1.$$

    (b)对于 $|x|<1$，推导出下面的式子：

$$1+x+x^2+x^3+\cdots=(1+x)(1+x^2)(1+x^4)(1+x^8)\cdots(1+x^{2^k})\cdots.$$

    (c)证明下面的结论：任意一个整数可以唯一地写成二进制形式，即如这样的和的形式：$a_0 2^0+a_1 2^1+a_2 2^2+\cdots$，其中 a_i 是 0 或 1(这一结论是构成计算机二进制运算的基础中非常重要的结论).
17. 确定下列等式/不等式的整数解的数量：
    (a)$b_1+b_2+b_3+b_4+b_5+b_6=15$，且对于所有的 i 有 $0\leqslant b_i\leqslant 3$.

(b)$b_1+b_2+b_3=15$，且对于所有 i 有 $0\leqslant b_i$，b_1 是奇数，b_2 是偶数，b_3 是素数.

(c)$b_1+b_2+b_3+b_4=20$，$2\leqslant b_1\leqslant 4$，且对于 $2\leqslant i\leqslant 4$ 有 $4\leqslant b_i\leqslant 7$.

(d)$b_1+b_2+b_3+b_4\leqslant 10$，且对于所有 i 有 $0\leqslant b_i$[提示：增加一个“松弛”变量 b_5 来创建一个等式].

18. 使用 2.10 节的分装问题方法求解例 5.20 的问题.
19. 使用例 5.22 的技术，寻找若干对 4 面骰子，使其与一对“标准”4 面骰子有同样的结果概率(“标准”4 面骰子的标签是 1，2，3，4).

5.4　二项式定理

为了扩展方程(5.19)的生成函数，求出函数 $f(x)=(1+x)^u$ 的麦克劳林级数是很有用的. 其中，u 是任意实数，正的或负的，没有必要一定是整数. 我们有

$$\begin{aligned} f'(x) &= u(1+x)^{u-1} \\ f''(x) &= u(u-1)(1+x)^{u-2} \\ &\vdots \\ f^{(r)}(x) &= u(u-1)\cdots(u-r+1)(1+x)^{u-r}. \end{aligned}$$

312　因此，根据等式(5.1)，我们有下面的定理.

定理 5.3(二项式定理)

$$(1+x)^u = 1+ux+\frac{u(u-1)}{2!}x^2+\cdots+\frac{u(u-1)\cdots(u-r+1)}{r!}x^r+\cdots. \tag{5.21}$$

可以证明，对于 $|x|<1$，展开式(5.21)成立. 通过引入下面的一般二项式系数(generalized binomial coefficient)可以简写这一展开式.

$$\binom{u}{r}=\begin{cases} \dfrac{u(u-1)\cdots(u-r+1)}{r!} & \text{如果 } r>0 \\ 1 & \text{如果 } r=0, \end{cases}$$

这一系数对任意实数 u 和非负整数 r 都有定义. 因此，(5.21)式可以重写成如下形式：

$$(1+x)^u = \sum_{r=0}^{\infty}\binom{u}{r}x^r. \tag{5.22}$$

如果 u 是正整数 n，那么对于 $r>u$，$\binom{u}{r}$ 等于 0，而且(5.22)式可简化成二项式展开(定理 2.7).

【例 5.25 计算平方根】 在返回到生成函数之前，让我们给出二项式定理对平方根计算的一个快速应用. 如果 $|x|<1$，我们有

$$(1+x)^{1/2} = 1+\frac{1}{2}x+\frac{\frac{1}{2}\left(\frac{1}{2}-1\right)}{2!}x^2+\frac{\frac{1}{2}\left(\frac{1}{2}-1\right)\left(\frac{1}{2}-2\right)}{3!}x^3+\cdots, \tag{5.23}$$

利用这一结果计算 $\sqrt{30}$. 注意到 $|29|\geqslant 1$，所以我们不能直接使用(5.23)式. 然而，因为

$$\sqrt{30}=\sqrt{25+5}=5\sqrt{1+0.2},$$

所以我们可以对 $x=0.2$ 使用(5.23)式. 这给出下面的结果：

$$\sqrt{30}=5\left[1+\frac{1}{2}(0.2)-\frac{1}{8}(0.2)^2+\frac{1}{16}(0.2)^3-\cdots\right]\approx 5.4775.$$ ■

返回到例 5.19，让我们运用二项式定理确定下面展开式中 x^k 的系数：

$$G(x)=(1+x+x^2+\cdots)^p.$$

注意，当 $|x|<1$ 时，使用恒等式(5.2)有

$$G(x)=\left(\frac{1}{1-x}\right)^{p},$$

$G(x)$可以重写成如下形式：

$$G(x)=(1-x)^{-p}.$$

313

现在，我们可以用$-x$取代x，且$u=-p$，然后使用二项式定理. 于是我们有

$$G(x)=\sum_{r=0}^{\infty}\binom{-p}{r}(-x)^{r}.$$

对于$k>0$，x^k的系数是

$$\binom{-p}{k}(-1)^{k}=\frac{(-p)(-p-1)\cdots(-p-k+1)}{k!}(-1)^{k},$$

这一系数等于

$$\begin{aligned}\frac{p(p+1)\cdots(p+k-1)}{k!}&=\frac{(p+k-1)(p+k-2)\cdots p}{k!}\\&=\frac{(p+k-1)!}{k!(p-1)!}=\binom{p+k-1}{k}.\end{aligned}$$

因为$\binom{p+0-1}{0}=1$，$\binom{p+k-1}{k}$也给出x^0的系数. 所以我们证明了下面的定理.

定理 5.4　如果存在p种类型的对象，且允许不限制每一种类型的重复，那么选取k个对象的不同方法数量由下式给出：

$$\binom{-p}{k}(-1)^{k}=\binom{p+k-1}{k}.$$

推论 5.4.1　假设p是一个固定的正整数，且c_k如下所示：

$$c_k=\binom{p+k-1}{k},$$

那么序列(c_k)的普通生成函数由下式给出：

$$C(x)=\left(\frac{1}{1-x}\right)^{p}=(1-x)^{-p}.$$

证明　这是上面证明的推论. ■

注意，$p=2$的情况包括在例 5.12 中. 在这一情况中，有

$$c_k=\binom{2+k-1}{k}=\binom{k+1}{k}=k+1.$$

注意，使用分装问题(5.3.2 节)的陈述方式，定理 5.4 说的是把k个不可区分的球放入到p个不同的盒子里，且对每个盒子里的球数不加以限制，存在的方法数量由下式给出：

$$\binom{p+k-1}{k}.$$

我们已在定理 2.4 中看到过这一结果.

314

【例 5.26 比萨饼问题(再探例 2.20)】　假设一家经营比萨饼的餐厅有九种类型的浇头(参见例 2.20). 如果一张比萨至多有一种浇头，有多少种方法可以卖 100 张比萨饼？实际上，可以合理地假设每一种浇头的供给不受限制. 现在我们有$p=10$个类型的浇头，包括浇头“除干酪没有其他浇头”. 那么根据定理 5.4，选取$k=100$张比萨的不同的方法数量由$\binom{109}{100}$给出. ■

【例 5.27 计算机系统评估】 以不同的标准对可选择的计算机系统评估，每一个标准有整数 1 到 6 的评估分. 有多少种方法可以使三个标准的总分数之和等于 12？为了回答这一问题，我们把每一标准上的分数看成是要选择的点数 1，2，3，4，5 或 6. 因此，要考虑的生成函数是

$$G(x) = (x + x^2 + \cdots + x^6)^3.$$

注意，我们开始于 x 而不是 1(或 x^0)，因为至少要选择一点. 我们取三次幂，因为有三个标准，我们想知道 x^{12} 的系数. 如何确定这一系数呢？这一结果要使用恒等式：

$$1 + x + x^2 + \cdots + x^s = \frac{1 - x^{s+1}}{1 - x}. \tag{5.24}$$

那么我们注意到

$$\begin{aligned} G(x) &= [x(1 + x + x^2 + \cdots + x^5)]^3 \\ &= x^3 \left[\frac{1 - x^6}{1 - x}\right]^3 \\ &= x^3(1 - x^6)^3(1 - x)^{-3}. \end{aligned} \tag{5.25}$$

我们已经知道 $C(x) = (1 - x)^{-p}$ 是序列 (c_k) 的生成函数，其中

$$c_k = \binom{p + k - 1}{k}.$$

这里

$$c_k = \binom{3 + k - 1}{k}.$$

使用二项式展开式(expansion)展开 $B(x) = x^3(1 - x^6)^3$，我们得到

315

$$B(x) = x^3[1 - 3x^6 + 3x^{12} - x^{18}] = x^3 - 3x^9 + 3x^{15} - x^{21}.$$

因此，$B(x)$ 是下面序列的生成函数：

$$(b_k) = (0,0,0,1,0,0,0,0,0,-3,0,0,0,0,0,3,0,0,0,0,0,-1,0,0,\cdots).$$

根据(5.25)式得到，$G(x)$ 是序列 (b_k) 和 (c_k) 的卷积的生成函数. 我们要确定 x^{12} 的系数 a^{12}. 这可以通过如下计算得到：

$$\begin{aligned} a_{12} &= b_0c_{12} + b_1c_{11} + b_2c_{10} + \cdots + b_{12}c_0 = b_3c_9 + b_9c_3 \\ &= 1 \cdot \binom{3 + 9 - 1}{9} - 3 \cdot \binom{3 + 3 - 1}{3} \\ &= \binom{11}{9} - 3\binom{5}{3} = 25. \end{aligned}$$

■

【例 5.28 列表 T 着色】 3.4 节的着色多项式被用于寻找使用固定数量的颜色着色给定图的可能的不同着色数量. 这里我们使用生成函数寻找无标签图 G 的不同列表 T 着色的数量(参见例 3.20 和例 3.22). G 的一个*列表 T 着色*(list T-coloring)是一个 T 着色，其中指定给顶点 x 的颜色属于与 x 相关的列表 $L(x)$. 我们将假设 $T = \{0, 1, 2, \cdots, r\}$，$G = K_n$，而且对于每一个顶点列表 $\{1, 2, \cdots, l\}$ 都是可能的颜色列表.

作为一个例子，假设 $r = 1$，$n = 5$，及 $l = 16$. 考虑使用颜色 1，3，5，8，10 的 K_5 的列表 T 着色. 每一对颜色的差的绝对值至少为 2，且有 $1 \leqslant 1 < 3 < 5 < 8 < 10 \leqslant 16$. 这组不等式给出一组差，即 $1-1$，$3-1$，$5-3$，$8-5$，$10-8$，$16-10$，或者是 0，2，2，3，2，6，且这些差的和等于 15. 注意，除了第一个差和最后一个差之外，所有的差都不等于 0 或 1，因为 $T = \{0, 1\}$ 而且 0 或 1 的差只能出现于第一个差(在这种情况下，需要一个 0)或最后一个差.

接下来，考虑另一个和等于 15 的非负整数序列，比如说，3，3，2，2，3，2. 这一序列可

以认为是下面不等式集合的差集合：$1\leqslant 4<7<9<11<14\leqslant 16$，因此认为是使用颜色 4，7，9，11，14 的 K_5 的一个列表 T 着色．注意，在这个五整数递增列表中，连续整数间的所有的绝对差都大于 1.

因此，很容易看到，要计数的这个列表 T 着色与下面方程的整数解之间是一一对应的：

$$b_1+b_2+b_3+b_4+b_5+b_6=15,$$

其中 b_1，$b_6\geqslant 0$，b_2，b_3，b_4，$b_5\geqslant 2$．因此，使用例 5.20 的想法，我们看到本例中的列表 T 着色的数量是下面函数中 x^{15} 的系数：

316

$$\begin{aligned}f(x)&=(1+x+x^2+\cdots)^2(x^2+x^3+\cdots)^4\\&=x^8(1-x)^{-6}.\end{aligned}$$

等价地，它是$(1-x)^{-6}$中 x^7 的系数．而这一 x^7 的系数等于

$$\binom{-6}{7}(-1)^7=\binom{6+7-1}{7}=\binom{12}{7}=792.$$

■

【例 5.29 弱序的数量】 我们可以使用生成函数计数 n 元素集合 X 上的不同弱序(X,R)的数量．回想一下，弱序有层次，而且每一层次上的元素是等价的．即如果 x，$y\in X$，那么，对于 x 的层次高于 y 的层次有 xRy，如果 x，y 处于同一层次，那么有 xRy 且 yRx．考虑有 w 个层次的弱序 R．每一个层次上至少存在一个元素．因此，带有 w 个层次的 n 集合上的弱序数量，是下面生成函数中 x^n 的系数：

$$\begin{aligned}G(x)&=\underbrace{(x+x^2+\cdots)(x+x^2+\cdots)\cdots(x+x^2+\cdots)}_{w\text{ 项}}\\&=(x+x^2+\cdots)^w\\&=x^w(1+x+x^2+\cdots)^w\\&=x^w(1-x)^{-w}.\end{aligned}$$

于是这是$(1-x)^{-w}$中 x^{n-w}的系数，即

$$\binom{-w}{n-w}(-1)^{n-w}=\binom{w+(n-w)-1}{n-w}=\binom{n-1}{n-w}.$$

因为层次数 w 的范围从 1 到 n，所以 n 集合上的不同弱序的数量总数等于

$$\binom{n-1}{n-1}+\binom{n-1}{n-2}+\cdots+\binom{n-1}{n-n}=\sum_{w=1}^{\infty}\binom{n-1}{n-w}.$$

根据定理 2.8[及帕斯卡三角形(2.7 节)]，其和等于 2^{n-1}.

■

本节练习

1. 使用二项式定理确定下列式子的展开式中 x^3 的系数：

   (a) $\sqrt[4]{1+x}$　　(b) $(1+x)^{-3}$

   (c) $(1-x)^{-4}$　　(d) $(1+5x)^{3/4}$

2. 确定下列式子的展开式中 x^7 的系数：

   (a) $(1-x)^{-4}x^3$　　(b) $(1-x)^{-2}x^8$

   (c) $x^2(1+x^2)^3(1-x)^{-3}$　　(d) $(1+x)^{1/2}x^5$

317

3. 如果$(1+x)^{1/3}$是序列(a_k)的普通生成函数，确定 a_2.
4. 做计算，求解 5.3 节练习 6(a).
5. 如果 a，b 是仅有的可用字母，使用定理 5.4 计算选出六个字母的方法数量．通过写出所有的方法检查你的答案.
6. 如果有 6 个不同厂家的计算机可选，那么有多少种方法可以选出 11 台计算机呢？

7. 如果四个不同公司中的每一个有 100 份股票可选，那么有多少种方法选出 50 份股票呢？
8. 果蝇因其眼睛的颜色被分类为显性的、混种的和隐性的. 选出 10 只果蝇做一个实验. 如果你只感兴趣显性、混种、隐性类型的果蝇数量，那么选择基因型(分类法)显性，混种和隐性有多少种不同的方法呢？
9. 五家不同的银行提供面值为 1000 美元整数倍的存款证(CD). 如果一位投资者有 10 000 美元，那么她有多少种不同的方法以 CD 形式投资？
10. 假设有六种不同的可用水果，(理论上)每一种水果的供给不受限制. 有多少种不同的装有 10 个水果的果篮？
11. 一个人每天晚上喝一罐啤酒，从六种不同品牌中选取一种. 假设给定品牌所提供的啤酒罐数不受限制，其中任意两罐是可交换的，而且我们不区分是在星期一喝 x 品牌的啤酒还是在星期二喝 x 品牌的啤酒. 那么在一周期间有多少种喝啤酒的不同方法？
12. 在研究一个工厂的次品时，我们把在一周内所发现的次品按其出现的日期分类. 有多少种不同的方法分类 10 个次品？
13. 假设有 p 种不同类型的对象，每一种对象的供给都不受限制. 如果我们必须从每一种类型至少选出一个对象，并设 a_k 是选出 k 个对象的不同方法数量.

   (a)构建 a_k 的生成函数.

   (b)序列(a_k)是生成函数为 x^p 的序列(b_k)和某个序列(c_k)的卷积，确定(c_k).

   (c)对所有的 k，确定 a_k.
14. 对于 5.3 节的练习 7，求 a_k. [提示：在生成函数中设 $y=x^2$.]
15. 假设 $B(x)$是序列(b_k)的普通生成函数. 设

$$Sb_k = b_0 + b_1 + \cdots + b_k,$$

   而且

318

$$S^2(b_k) = S(Sb_k) = \sum_{j=0}^{\infty}(b_0 + b_1 + \cdots + b_j).$$

   一般地，设 $a_k = S^p(b_k) = S(S^{p-1}(b_k))$. 那么我们可以证明

$$a_k = b_k + pb_{k-1} + \cdots + \binom{p+j-1}{j}b_{k-j} + \cdots + \binom{p+k-1}{k}b_0.$$

   (a)对于 $p=2$，证明上述表达式.

   (b)如果 $A(x)$是序列(a_k)的普通生成函数，利用 $B(x)$确定 $A(x)$的表达式.
16. 考虑例 5.28 的列表 T 着色问题.

   (a)当 r，n，l 分别取下列各组值时，求解这一问题：

   i. $r=1$，$n=5$ 和 $l=22$

   ii. $r=1$，$n=4$ 和 $l=16$

   iii. $r=2$，$n=5$ 和 $l=16$

   (b)对于任意正整数 r，n 和 l，确定一般解.
17. 考虑例 5.29 中的弱序计数问题. 10 元素集合上有多少种不同的弱序？其中：

   (a)4 个元素处于最高层次.

   (b)最多有 4 个元素处于最高层次.

   (c)每一个层次上有偶数个元素.
18. 设 p_n^r 是把整数 n 正好划分成 r 个部分的数量，并考虑划分的顺序. 例如，考虑顺序时，有 10 种方法把 6 正好划分成 4 个部分，即

$$\{3,1,1,1\},\quad \{1,3,1,1\},\quad \{1,1,3,1\},\quad \{1,1,1,3\},\quad \{2,2,1,1\},$$
$$\{2,1,2,1\},\quad \{2,1,1,2\},\quad \{1,2,2,1\},\quad \{1,2,1,2\},\quad \{1,1,2,2\}.$$

   (a)构建 p_n^r 的普通生成函数.

   (b)求 p_n^r.

19. 三个变量 u, v, w 的多项式称为是齐次的(homogeneous)，如果各项 $au^iv^jw^k$ 的总阶数相同，即如果 $i+j+k$ 是一个常数. 例如，

$$3v^4+2uv^2w+4vw^3$$

就是齐次的，其中每一项的总阶数为 4. 一个各项总阶数为 n 的三变量齐次多项式中可能的最大项数是多少?

20. (a)证明练习 18 中定义的 p_n^r 是各项总阶数为 n，且每一项中各变量的阶至少为 1 的 r 个变量齐次多项式的最大可能项数.

(b)使用上述结果和练习 18 的结果回答练习 19 的问题.

21. 三个人每一个人投一次骰子. 有多少方法可以使总点数之和等于 9? [319]

5.5 指数生成函数和排列的生成函数

5.5.1 指数生成函数的定义

迄今为止，我们使用了普通生成函数计数对象组合的数量，我们使用术语组合(combination)，是因为不考虑对象的顺序. 现在我们尝试在考虑对象的顺序时做类似的工作并计数排列. 回想一下，$P(n, k)$是 n 集合的 k 排序的数量. 固定 n，$P(n, k)$的普通生成函数由下式给出:

$$\begin{aligned}G(x)&=P(n,0)x^0+P(n,1)x^1+P(n,2)x^2+\cdots+P(n,n)x^n\\&=\sum_{k=0}^{\infty}\frac{n!}{(n-k)!}x^k.\end{aligned}$$

遗憾的是，没有化简这一表达式的好方法. 假如我们研究过组合，以及从 n 集合中取 k 个元素的方法数量 $C(n, k)$，那么我们就能够化简，因为我们有下面的表达式:

$$C(n,0)x^0+C(n,1)x^1+C(n,2)x^2+\cdots+C(n,n)x^n, \tag{5.26}$$

通过二项式展开，上述展开式可以化简成$(1+x)^n$. 根据定理 2.1，有

$$P(n,r)=C(n,r)P(r,r)=C(n,r)r!.$$

因此，$(1+x)^n$ 与(5.26)式等价可以重写成

$$P(n,0)\frac{x^0}{0!}+P(n,1)\frac{x^1}{1!}+P(n,2)\frac{x^2}{2!}+\cdots+P(n,n)\frac{x^n}{n!}=(1+x)^n. \tag{5.27}$$

数 $P(n, k)$是$(1+x)^n$ 的展开式中 $x^k/k!$ 的系数.

这暗示下面的想法. 如果(a_k)是任意序列，这一序列的**指数生成函数**(exponential generating function)是

$$H(x)=a_0\frac{x^0}{0!}+a_1\frac{x^1}{1!}+a_2\frac{x^2}{2!}+\cdots+a_k\frac{x^k}{k!}+\cdots=\sum_k a_k\frac{x^k}{k!}.$$

与普通生成函数一样，我们考虑使上述表达式收敛的 x⊖. [320]

作为一个例子，如果对于 $k=0$, 1, …, $a_k=1$，那么利用等式(5.3)，我们看到它的指数生成函数是

$$H(x)=1\cdot\frac{x^0}{0!}+1\cdot\frac{x^1}{1!}+1\cdot\frac{x^2}{2!}+\cdots=1+\frac{x^1}{1!}+\frac{x^2}{2!}+\cdots=e^x.$$

作为另一个例子，如果 $a_k=P(n, k)$，那么我们已在(5.27)式中证明了它的指数生成函数是$(1+x)^n$. 再给出一个例子，假设 α 是任意实数，且(a_k)是序列$(1, \alpha^1, \alpha^2, \alpha^3, \cdots)$. 那么$(a_k)$的指数生成函数是

⊖ 如 P190 的脚注所提示的那样，我们可以使用形式幂级数精确地定义这个概念.

$$H(x)=\sum_{k=0}^{\infty}a^k\frac{x^k}{k!}=\sum_{k=0}^{\infty}\frac{(ax)^k}{k!}=e^{ax}.$$

同普通生成函数的情况一样，我们将往返于序列与指数生成函数之间.

【例 5.30 欧拉图】 一个连通图称为欧拉的(eulerian)，如果每一个顶点有偶数度. 欧拉图在第 11 章讨论的各种应用中很重要. Harary 和 Palmer[1973]以及 Read[1962]都指出，如果 u_n 是 n 顶点标签连通欧拉图的数量，那么序列(u_n)的指数生成函数 $U(x)$由下式给出：

$$U(x)=x+\frac{x^3}{3!}+\frac{3x^4}{4!}+\frac{38x^5}{5!}+\cdots.$$

因此，如图 5.7 所示，存在一个三顶点标签连通欧拉图，以及三个四顶点标签连通欧拉图.

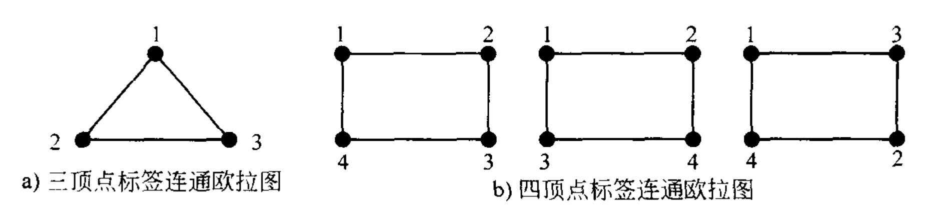


图　5.7

■

5.5.2　排列计数的应用

【例 5.31】 一个代码使用三个不同的字母 a，b 或 c. 五个或更少的字母序列给出一个代码字. 这一代码字可以使用至多一个 b、至多一个 c 和至多三个 a. 长度为 k 且 $k\leqslant 5$ 的所有可能的代码字有多少? 注意，在代码字中要考虑顺序. 例如，代码字 aab 和 aba 是不同的，尽管之前考虑子集时，$\{a, a, b\}$和$\{a, b, a\}$是相同的. 所以，我们感兴趣的是计数排列而不是组合. 然而，根据我们之前的经验，我们以计数组合数开始：在至多选一个 b，至多选一个 c 且至多选三个 a 的前提下计数得到 k 个字母的方法数. 普通生成函数是由下面的式子计算而来的：

321

$$(1+ax+a^2x^2+a^3x^3)(1+bx)(1+cx),$$

它等于

$$1+(a+b+c)x+(bc+a^2+ab+ac)x^2+(a^3+abc+a^2b+a^2c)x^3$$
$$+(a^2bc+a^3b+a^3c)x^4+a^3bcx^5.$$

其中 x^k 的系数给出得到 k 个字母的方法. 例如，选出三个字母情况如下：3 个 a，a 和 b 和 c，2 个 a 和 b，2 个 a 和 c. 当我们选择其中的一种情况 a 和 b 和 c 时，有 3! 个所对应的排列：

$$abc, acb, bac, bca, cab, cba.$$

对于 3 个 a 的选择，则只有一种对应的排列：aaa. 对于 2 个 a 和 b 的选择，有 3 种排列：aab，aba，baa. 根据定理 2.6 的一般公式，我们知道为什么这是成立的. 三个对象中有 2 个是一种类型，另外一个对象是一种类型时，所有不同排列数量由下式给出：

$$\frac{3!}{2!1!}.$$

一般地，如果我们有 n_1 个 a，n_2 个 b 和 n_3 个 c，那么所对应的排列数量是

$$\frac{n!}{n_1!n_2!n_3!}.$$

特别地，在我们的讨论中，如果选择三个字母，那么得到码字的正确信息的方法由下式给出：

322

$$\frac{3!}{3!}a^3+\frac{3!}{1!1!1!}abc+\frac{3!}{2!1!}a^2b+\frac{3!}{2!1!}a^2c. \tag{5.28}$$

令 $a=b=c=1$ 将会产生正确的计数这样的三字母码字的数量. 使用下面的技巧

$$\frac{(ax)^p}{p!}=\frac{a^p}{p!}x^p$$

取代 a^px^p 来得到我们的示意生成函数，可以得到(5.28)式以及其他适当的系数.

在我们的例子中有：

$$\left(1+\frac{a}{1!}x+\frac{a^2}{2!}x^2+\frac{a^3}{3!}x^3\right)\left(1+\frac{b}{1!}x\right)\left(1+\frac{c}{1!}x\right),$$

这等于

$$1+\left(\frac{a}{1!}+\frac{b}{1!}+\frac{c}{1!}\right)x+\left(\frac{bc}{1!1!}+\frac{a^2}{2!}+\frac{ab}{1!1!}+\frac{ac}{1!1!}\right)x^2+$$
$$\left(\frac{a^3}{3!}+\frac{abc}{1!1!1!}+\frac{a^2b}{2!1!}+\frac{a^2c}{2!1!}\right)x^3+\left(\frac{a^2bc}{2!1!1!}+\frac{a^3b}{3!1!}+\frac{a^3c}{3!1!}\right)x^4+\frac{a^3bc}{3!1!1!}x^5. \tag{5.29}$$

把 x^3 的系数和(5.28)式中的表达式做比较，上面的表达式仍然不令人满意. 然而，如果我们把这一表达式考虑成指数生成函数并选择 $x^k/k!$ 的系数，那么这一表达式就很好了. 因为表达式(5.29)等于

$$1+1!\left(\frac{a}{1!}+\frac{b}{1!}+\frac{c}{1!}\right)\frac{x}{1!}+2!\left(\frac{bc}{1!1!}+\frac{a^2}{2!}+\frac{ab}{1!1!}+\frac{ac}{1!1!}\right)\frac{x^2}{2!}+$$
$$3!\left(\frac{a^3}{3!}+\frac{abc}{1!1!1!}+\frac{a^2b}{2!1!}+\frac{a^2c}{2!1!}\right)\frac{x^3}{3!}+\cdots. \tag{5.30}$$

令 $a=b=c=1$ 且取 $x^k/k!$ 的系数给出适当的代码字(排列)数量. 例如，长度为 3 的代码字数量是

$$3!\left(\frac{1}{3!}+1+\frac{1}{2!}+\frac{1}{2!}\right)=13.$$

所对应的代码字是我们已经列出的六种有一个 a、一个 b 和一个 c 的情况，以及三种有 2 个 a 和 1 个 b 的情况，三种有 2 个 a 和一个 c 的情况和一种有 3 个 a 的情况. ■

例 5.31 的分析扩展成如下定理.

定理 5.5　假设我们有 p 种类型的对象，且对于 $i=1, 2, \cdots, p$，类型 i 有 n_i 个不可区分的对象. 长度为 k 且类型 i 最多有 n_i 个对象的不同排列的数量，是下面指数生成函数中 $x^k/k!$ 的系数：

$$\left(1+x+\frac{x^2}{2!}+\cdots+\frac{x^{n_1}}{n_1!}\right)\left(1+x+\frac{x^2}{2!}+\cdots+\frac{x^{n_2}}{n_2!}\right)\cdots\left(1+x+\frac{x^2}{2!}+\cdots+\frac{x^{n_p}}{n_p!}\right).$$

323

【例 5.32 RNA 链】　作为上述结果的应用，考虑当我们最多可以使用 3 个 A、3 个 G、2 个 C 和 1 个 U 时，2 链接 RNA 链的数量. 因为要考虑顺序，我们寻找一个指数生成函数. 这一指数生成函数由下式给出：

$$\left(1+x+\frac{x^2}{2!}+\frac{x^3}{3!}\right)^2\left(1+x+\frac{x^2}{2!}\right)(1+x),$$

上式其实等于

$$1+4x+\frac{15}{2}x^2+\frac{53}{6}x^3+\cdots.$$

这里 x^2 的系数是 15/2，所以 $x^2/2!$ 的系数是 $2!(15/2)=15$. 因此，有 15 个这样的链. 它们是 AA，AG，AC，AU，GA，GG，GC，GU，CA，CG，CC，CU，UA，UG 和 UC，即除 *UU* 之外的所有 2 链接 RNA 链. 类似地，这些可能的基组成的 3 链接 RNA 链的数量是 $x^3/3!$，或 $3!(53/6)=53$. 读者应该能够检验这一结果. ■

【例 5.33 RNA 链(继续)】　继续例 5.32，假设每一个基都可以提供任意数量，我们希望寻找

长度为 k 的 RNA 链的数量．其指数生成函数由下式给出：

$$\left(1+x+\frac{x^2}{2!}+\frac{x^3}{3!}+\cdots\right)^4=(e^x)^4=e^{4x}=\sum_{k=0}^{\infty}\frac{(4x)^k}{k!}=\sum_{k=0}^{\infty}4^k\frac{x^k}{k!}.$$

因此，所讨论的 RNA 链的数量是 4^k．这与在第 2 章中只使用乘法规则所得的结论一致．

在这里，我们做一个修改，即计数 U 链接数为偶数时长度为 k 的 RNA 链的数量．指数生成函数由下式给出：

$$H(x)=\left(1+\frac{x^2}{2!}+\frac{x^4}{4!}+\cdots\right)\left(1+x+\frac{x^2}{2!}+\cdots\right)^3.$$

现在，$H(x)$中的第二项由$(e^x)^3=e^{3x}$给出．不难证明 $H(x)$中的第一项由下式给出：

324

$$\frac{1}{2}(e^x+e^{-x}).$$

因此，有

$$H(x)=\frac{1}{2}(e^x+e^{-x})(e^{3x})=\frac{1}{2}(e^{4x}+e^{2x})$$
$$=\frac{1}{2}\left[\sum_{k=0}^{\infty}4^k\frac{x^k}{k!}+\sum_{k=0}^{\infty}2^k\frac{x^k}{k!}\right]=\sum_{k=0}^{\infty}\left(\frac{4^k+2^k}{2}\right)\frac{x^k}{k!}.$$

我们得出所讨论的 RNA 链的数量是

$$\frac{4^k+2^k}{2}.$$

为了检验这一结果，注意，如果 $k=2$，那么这个数是 10．这 10 个链是 UU，GG，GA，GC，AG，AA，AC，CG，CA 和 CC． ■

5.5.3 可区分球到不可区分盒子的分配㊀

回顾 2.10.4 节的内容，第二类斯特林数 $S(n,k)$的定义为把 n 个可区分球分配到 k 个可区分盒子里，且每一个盒子都不为空时的方法数量．这里，我们证明

$$S(n,k)=\frac{1}{k!}\sum_{i=0}^{k}(-1)^i\binom{k}{i}(k-i)^n. \tag{5.31}$$

首先，我们考虑确定把 n 个可区分球放入分别标有标签 1，2，…，k 的可区分盒子里，且每一个盒子都不为空的方法数量 $T(n,k)$的问题．注意

$$T(n,k)=k!S(n,k), \tag{5.32}$$

因为我们是通过下面的方法得到把 n 个可区分球放入 k 个不为空的可区分盒子的分配的：首先确定把 n 个可区分球放入 k 个不为空的不可区分盒子的分配，然后再标记(排序)这些盒子．接下来，计算 $T(n,k)$．假设球 i 进入到盒子 $C(i)$．我们可以通过给出序列 $C(1)C(2)\cdots C(n)$编码把球放入可区分盒子里的分配．这是 k 集合$\{1,2,\cdots,k\}$的一个 n 排列，且 k 集合中的每一个标签 j 至少使用一次．因此，$T(n,k)$是这一排列的数量，且对于固定的 k，$T(n,k)$的指数生成函数由下式给出：

325

$$H(x)=\left(x+\frac{x^2}{2!}+\frac{x^3}{3!}+\cdots\right)^k=(e^x-1)^k.$$

$T(n,k)$是 $H(x)$的展开式中 $x^n/n!$ 的系数．根据二项式展开(定理 2.7)，有

$$H(x)=\sum_{i=0}^{k}\binom{k}{i}(-1)^i e^{(k-i)x}.$$

㊀ 本小节可以跳过．

在(5.3)式的 e^x 的幂级数中，用$(k-i)x$取代x，我们得到

$$H(x)=\sum_{i=0}^{k}\binom{k}{i}(-1)^i\sum_{n=0}^{\infty}\frac{1}{n!}(k-i)^n x^n=\sum_{n=0}^{\infty}\frac{x^n}{n!}\sum_{i=0}^{k}(-1)^i\binom{k}{i}(k-i)^n.$$

确定 $H(x)$的展开式中 $x^n/n!$ 的系数，我们有

$$T(n,k)=\sum_{i=0}^{k}(-1)^i\binom{k}{i}(k-i)^n. \tag{5.33}$$

现在方程(5.32)和方程(5.33)给出方程(5.31).

本节练习

1. 对于下列每一个序列(a_k)，求其指数生成函数的简单、闭式的表达式.

   (a)(5，5，5，…) (b)$a_k=3^k$ (c)(1，0，0，1，1，…)

   (d)(0，0，1，1，…) (e)(1，0，1，0，1，…) (f)(2，1，2，1，2，1，…)

2. 对于下列每一个函数，寻找一个序列，使这一函数是该序列的指数生成函数.

   (a)$4+4x+4x^2+4x^3+\cdots$ (b)$\frac{3}{1-x}$ (c)x^2+5e^x

   (d)$x^2+4x^3+x^5$ (e)e^{6x} (f)$5e^x$

   (g)$e^{2x}+e^{5x}$ (h)$(1+x^2)^n$ (i)$\frac{1}{1-6x}$

3. 一个图被称为偶性的(even)，如果每一个顶点有偶数度. 令 e_k 是k 个顶点标签偶性图的数量，Harary 和 Palmer[1973]证明序列(e_k)的指数生成函数 $E(x)$由下式给出：

$$E(x)=x+\frac{x^2}{2!}+\frac{2x^3}{3!}+\frac{8x^4}{4!}+\cdots.$$

   验证 $x^3/3!$ 和 $x^4/4!$ 的系数.(注意，e_k 可以由 11.3 节的练习 11 和 3.1.3 节的结果得到.)

4. 对于例 5.32，通过枚举法验证存在由可能的基组成的 53 个 3 链接 RNA 链. [326]

5. 如果可能基是 2 个 A、3 个 G、3 个 C 和 1 个 U. 求 3 链接 RNA 链的数量.

6. 对于下列每一种情况，构建适当的生成函数，但不必计算出答案. 指出你在寻找什么，例如，x^8 的系数.

   (a)当用a，b，c 和d 构建三字母的代码字且b 和d 只能选择一次时，有多少个这样的代码字?

   (b)由 0、1、2、3 和 4 组成的长度为 6 的代码字，至少包含一个 0、一个 1、一个 2、一个 3 和一个 4. 这样的代码字有多少个?

   (c)至多有 4 个 0、3 个 1、4 个 2 的 11 位数字有多少个?

   (d)有多少种方法可以从 $2n$ 个A、$2n$ 个B 和 $2n$ 个C 中选出 $3n$ 个字母?

   (e)对于固定的偶数n，从数字表{0，1，2，3}生成n位数字的字，且每一个字中 0 的个数和 1 的个数是偶数而 2 的个数是奇数. 确定这样的字的数量.

   (f)如果投掷 50 次骰子，有多少种方法可以得到总点数 100?

   (g)10 种市政债券被分为 A，AA 或 AAA 三个等级. 有多少种指定这些等级的不同方法?

   (h)在计划一个为期 20 天的工作时间表时，有多少种方法可以在这 20 天中安排至多 5 个假日、至多 5 天私人时间和至多 15 个工作日?

   (i)假设在抛类型A的硬币时，如果正面向上则你得 1 点，如果反面向上则你得 2 点. 在抛类型B的硬币时，正面向上得 2 点，反面向上得 3 点. 如果你抛 3 枚A类型硬币和 5 枚B类型硬币，那么你有多少种方法得到 12 点?

   (j)如果在四个计算机机房中分配 200 台相同的终端，那么有多少种方法使得每一个计算机机房将得到 20 台或 40 台或 60 台或 80 台或 100 台终端?

   (k)一艘船与其他船进行可视联络的方法之一是在旗杆上挂起一系列各颜色的旗帜. 信号的意义依赖于从顶到下的旗帜的顺序. 如果有 5 面红色旗帜、4 面绿色旗帜、4 面黄色的旗帜和 1 面蓝色的旗帜可用，那么使用 12 面旗帜时可能有多少种不同的信号?

(l)在(k)中，如果至少使用 12 面旗帜，那么有多少种可能的信号？

7. (a)确定长度为 k 且 A 的数量是奇数的 RNA 链的数量.

   (b)对 $k=2$ 的情况进行说明.

8. 确定长度为 2 且 U 的数量是偶数或 A 的数量是奇数的 RNA 链的数量.

9. 假设存在 p 种不同的对象，每一种对象的供给不受限制. 设 a_k 是从这些对象中选出 k 个对象的排列数量. 使用指数生成函数显式确定 a_k.

327

10. 如果在 2 个计算机机房中分配 60 台同样的终端，那么有多少种方法使得每一个计算机机房有 20 台或 40 台终端？

11. 如果考虑顺序，确定整数 k 的划分数量的指数生成函数(5.4 节例 5.23 和练习 18).

12. 如果我们有 p 种类型的对象，且每一种类型的对象的供给不受限制，我们希望选出 k 个对象，且每一种类型至少有一个对象，考虑顺序. 确定选择数量的指数生成函数的一个简单、闭式表达式.

13. 如果我们有 p 种类型的对象，且每一种类型的对象的供给不受限制，我们希望选出 k 个对象，且每一种对象的选择数量是偶数(包括 0)，考虑顺序. 确定选择数量的指数生成函数的一个简单、闭式表达式.

14. 从字母表$\{a, b, c, d, e\}$中生成长度为 k 且 b 的出现次数为奇数次的代码字. 确定这种代码字的数量.

15. 确定从数字表$\{1, 2, 3, 4, 5, 6\}$中生成长度为 3 且 1，3，4 和 6 的出现次数为偶数次的代码字的数量.

16. 分别根据(5.31)式和(5.33)式计算 $S(4, 2)$和 $T(4, 2)$，并通过列出所有适当的分配来验证你的答案.

17. 练习 17 至 20 研究指数生成函数的组合. 假设 $A(x)$和 $B(x)$分别是序列(a_k)和(b_k)的指数生成函数. 确定指数生成函数为 $C(x)=A(x)+B(x)$的序列(c_k)的第 k 项 c_k 的表达式.

18. 对于 $C(x)=A(x)B(x)$，重复练习 17.

19. 如果(a_k)的指数生成函数是下列函数，确定 a_3.

    (a)$e^x(1+x)^6$ (b)$\frac{e^{3x}}{1-x}$ (c)$\frac{x^2}{(1-x)^2}$

20. 假设 $a_{n+1}=(n+1)b_n$，且 $a_0=b_0=1$. 如果 $A(x)$是序列(a_n)的指数生成函数，$B(x)$是序列(b_n)的指数生成函数，推导 $A(x)$和 $B(x)$之间的关系.

5.6 概率生成函数⊖

生成函数的简单思想在概率的研究中有着有趣的应用. 事实上，拉普拉斯在他的《Théorie Analytique des Probabilités》(概率的分析理论，Paris，1812)一书中首先对生成函数做了全面的研究，而且生成函数的研究动机大部分来自于概率. 假设在一次实验之后，我们知道将出现一个且仅一个可能事件的(有限或可数无限)集合. 设 p_k 是第 k 个事件出现的概率，其中 $k=0, 1, 2, \cdots$. (当然，如果存在可能事件的闭联集，那么这一表记法是不可行的.)下面的普通生成函数

328

$$G(x)=\sum p_k x^k \tag{5.34}$$

称为**概率生成函数**(probability generating function)[注意(5.34)式至少对 $|x|\leqslant 1$ 是收敛的，因为 $p_0+p_1+\cdots+p_k=1$]. 我们将看到概率生成函数在实验的评估，特别是在粗略地分析我们期待的结果的评估中非常有用.

【例 5.34 投掷硬币】 假设这一实验是投掷一枚公正的硬币. 那么事件是正面(H)和反面(T)，正面出现的概率 p_0 等于 1/2，反面出现的概率 p_1 等于 1/2. 因此，概率生成函数是

$$G(x)=\frac{1}{2}+\frac{1}{2}x.$$ ■

⊖ 跳过本节也不会失去连续性. 尽管本质上本节是独立的，但是对概率论有一些了解的读者，至少在诸如 Goodman 和 Ratti[1992]或 Kemeny，Snell 和 Thompson[1974]的“有限数学”的水平上的读者可以对此内容有更多的领会.

【例 5.35 伯努利试验】 在伯努利试验中，一个实验被独立地重复试验 n 次，每一次试验有成功概率 p，失败概率 $q=1-p$. 这一实验可以是检查一种产品是次品或正品的测试，或者是一种疾病存在或不存在的测试，或是决定是否接受或拒绝工作候选人. 如果 S 表示成功，F 表示失败，那么在一个 $n=5$ 的试验中，一个典型的结果是如 SSFSF 或 SSFFF 这样的序列. 在 n 次试验中存在 k 次成功的概率由下式给出：

$$b(k,n,p)=C(n,k)p^kq^{n-k},$$

正如任意一本关于概率论(诸如 Feller[1968]、Parzen[1992]、Ross[1997])，或关于有限数学(例如 Goodman 和 Ratti[1992]、Kemeny，Snell 和 Thompson[1974])的标准书籍中所论述的那样. n 次试验中成功数量的概率生成函数由下式给出：

$$G(x)=\sum_{k=0}^{n}b(k,n,p)x^k=\sum_{k=0}^{n}C(n,k)p^kq^{n-k}x^k.$$

根据二项式展开(定理 2.7)，我们有

$$G(x)=(px+q)^n.$$ ■

下面给出若干概率生成函数的简单结果.

定理 5.6 *如果 G 是概率生成函数，那么*

$$G(1)=1.$$

证明 因为根据假设，结果是相互排斥且是穷举的，所以我们有

$$p_0+p_1+\cdots+p_k+\cdots=1.$$ ■

329

推论 5.6.1

$$\sum_{k=0}^{n}C(n,k)p^kq^{n-k}=1.$$

证明 在伯努利试验中(例 5.35)，令 $G(1)=1$. ■

注意到$(p+q)n=1^n$ 这一结果，推论 5.6.1 也可以直接通过二项式展开证明.

假设在一项实验中，如果第 k 项事件出现，那么我们得到 k 美元(或 k 个单位的某种报酬). 那么表达式 $E=\sum kp_k$ 称为*期望值*(expected value)或*期望*(expectation). 这是当这一项实验被重复多次时我们期望"赢得"的平均值，我们期望在整体的 p_0 有机会获 0 美元、在 p_1 有机会获 1 美元，以此类推. 关于期望值的更详细讨论，可以参见任何一本关于概率论的初级书籍，或关于有限数学的书籍. 注意，只有当和$\sum kp_k$ 收敛时，期望值才有定义. 如果这一和的确收敛，我们说这一期望值*存在*(exist). 我们在总给出 1 美元的实验中与以 1/2 的概率给出 0 美元并以 1/2 的概率给出 2 美元的实验中得到相同的期望值. 然而，在第二个实验的结果中有更多的偏差. 概率学家引入了方差的概念来度量这一偏差. 特别地，*方差*(variance)V 定义为

$$V=\sum_k k^2p_k-\left(\sum_k kp_k\right)^2. \tag{5.35}$$

对于这一概念的详细解释可以参见诸如 Feller[1968]、Parzen[1992]或 Ross[1997]等概率论书籍. 只有当(5.35)式的和收敛时，方差才有定义. 在确实收敛的情况下，我们说方差*存在*. 在上面提到的第一个实验中，有

$$V=[1^2(1)]-[1(1)]^2=0.$$

在上面提到的第二个实验中，有

$$V=\left[0^2\left(\frac{1}{2}\right)+2^2\left(\frac{1}{2}\right)\right]-\left[0\left(\frac{1}{2}\right)+2\left(\frac{1}{2}\right)\right]^2=1.$$

因此，第二个实验的方差更大. 我们将看到概率生成函数是如何使我们得以计算期望值和方差的.

对 x 求(5.34)式的微分产生下面的等式：

330

$$G'(x)=\sum kp_kx^{k-1}.$$

因此，如果 $G'(x)$对于 $x=1$ 收敛，即如果$\sum kp_k$ 收敛，那么有

$$G'(1)=\sum kp_k. \tag{5.36}$$

如果第 k 个事件给出值 k 美元或 k 个单位报酬，那么(5.36)式右边的表达式是期望值.

定理 5.7 *假设 $G(x)$是概率生成函数且第 k 个事件给出值 k. 如果期望值存在，那么 $G'(1)$ 就是这一期望值.*

把定理 5.7 运用到伯努利试验中. 我们有

$$\begin{aligned} G(x) &=(px+q)^n \\ G'(x) &=n(px+q)^{n-1}p \\ G'(1) &=np(p+q)^{n-1} = np(1)^{n-1} = np. \end{aligned}$$

因此，在 n 次试验中，成功的期望数量是 np. 回想起这一事实的"标准"推导的读者应该对如此简单的推导感到高兴. 为了解释这一结果，我们注意到做 $n=100$ 次均匀的硬币投掷时，正面(成功)的概率是 $p=0.5$，而正面的期望数量是 $np=50$.

【例 5.36 芯片制造】 一家制造计算机芯片的公司估测在每 10 000 枚制造的芯片中有一枚是次品. 如果有 100 000 枚芯片的订单，那么对这一订单，期望的次品枚数是多少？假设次品的出现是独立的，那么这是一个伯努利试验的例子，而且我们看到这个期望值是(100 000)1/10 000=10. ■

【例 5.37 数据包传输】 在数据传输中，传输的"数据包"丢失的概率是 1000 分之 1. 在丢失数据包之前传输的数据包的期望数量是多少？假设数据包丢失是独立的，那么这是伯努利试验. 我们可以尝试首先计算*首次成功*(first success)出现在第 k 次试验的概率，然后我们计算首次成功的期望值. 我们要求读者在练习 7 中对此进行探讨. ■

下面的定理与方差有关. 它的证明留给读者(练习 9).

定理 5.8 *假设 $G(x)$是概率生成函数且第 k 个事件有值 k. 如果方差 V 存在，那么 $V=G''(1)+G'(1)-[G'(1)]^2$.*

把定理 5.8 运用于伯努利试验，我们有

331

$$G''(x) = n(n-1)p^2(px+q)^{n-2}.$$

且有

$$\begin{aligned} G'(1) &=np \\ G''(1) &=n(n-1)p^2. \end{aligned}$$

因此，有

$$\begin{aligned} V &=G''(1)+G'(1)-[G'(1)]^2 \\ &=n(n-1)p^2+np-n^2p^2 \\ &=np-np^2 = np(1-p) \\ &=npq. \end{aligned}$$

这给出这一方差的一个最著名公式 npq.

本节练习

1. 对于下列每一种情况，确定其概率生成函数的简单、闭式的表达式，并使用这一表达式计算期望值和

方差.

(a) $p_0=p_1=p_2=1/3$，对于其他的 k，$p_k=0$.

(b) $p_4=2/5$，$p_7=3/5$，对于其他的 k，$p_k=0$.

(c) $p_1=1/4$，$p_2=1/4$，$p_4=1/2$，对于其他的 k，$p_k=0$.

2. 制造小型发动机的一家公司估测在每 100 000 台制造的发动机中有两台次品. 如果有一个 100 台发动机的订单，那么对于这一订单，次品发动机的期望数量是多少?

3. 某种疾病被认为是非传染的. 一名研究人员估测这种疾病会在每 50 个人中发现一例. 这名研究者在找到一例病人之前必须检查的人的期望值是多少? 方差是多少?

4. 对于固定的正数 λ，带有参数 λ 的泊松分布(poisson distribution)有：

$$p_k=\frac{e^{-\lambda}\lambda^k}{k!},\quad k=0,1,2,\cdots.$$

(a)确定上述分布的概率生成函数的一个简单、闭式的表达式.

(b)使用生成函数的方法确定期望值和方差.

5. (Daniel[1995])Gibbons，Clark 和 Fawcett[1990]研究了在伊利诺伊州库可郡在 1977 年到 1987 年之间青春期自杀的月分布. 他们发现这一分布非常接近参数为 $\lambda=2.75$ 的泊松分布.

332

(a)如果随机选定一个月，确定有 4 位青春期自杀者的概率.

(b)确定每一个月自杀者的期望数量.

(c)确定其方差.

6. (Daniel[1995])假设从一个池塘选出大量的样本，且在一个样本中发现给定种类的水生有机物的平均数是 2. 假设水生有机物的数量满足泊松分布，确定下一次取样有 3 个或 4 个这样的有机物的概率.

7. 在伯努利试验中，假设我们计算首次成功出现在第 k 次试验的概率. 这一概率是：$k=0$ 时，$p_k=0$(假设我们从试验 1 开始)，$k>0$ 时，$p_k=(1-p)^{k-1}p$. 这些概率 p_k 定义几何分布(geometric distribution). 对于这一分布重复练习 4 并把结果运用于例 5.37 中的问题.

8. 固定一个正整数 m. 在伯努利试验中，在 $k+m$ 次试验时出现第 m 次成功的概率由下式给出：

$$p_k=\binom{k+m-1}{k}q^kp^m.$$

概率 p_k 定义负二项分布(negative binomial distribution).

(a)证明：负二项分布 p_k 的概率生成函数 $G(x)$ 由下式给出：

$$G(x)=\frac{p^m}{(1-qx)^m}.$$

(b)计算期望值和方差.

9. 证明定理 5.8.

5.7 Coleman 和 Banzhaf 的势力指数[⊖]

在 2.15 节，我们介绍了简单游戏及 Shapley-Shubik 势力指数的概念. 本节，我们将定义另外两个势力指数并讨论如何使用生成函数计算这两个势力指数. 对于联合 S，当 S 是赢联合时，我们定义 S 的值 $v(S)$ 为 1，当 S 是输联合时，我们定义 S 的值 $v(S)$ 为 0. Coleman[1971]定义玩家 i 的势力为

$$P_i^C=\frac{\sum_S[v(S)-v(S-\{i\})]}{\sum_S v(S)}.\tag{5.37}$$

在计算这一度量时，你可以对所有联合 S 求和. 如果从 S 去除 i 使其从赢联合变成输联合，那么项

⊖ 本节可以跳过而不失连续性.

$$v(S)-v(S-\{i\})$$

等于 1，否则这个项为 0(这个值不可能是 -1，因为我们假设了一个赢联合从不包含于一个输联

333

合中). 因此，P_i^C 是这样的数：从该赢联合中去除 i 导致一个输联合的赢联合数量除以全体赢联合数量，或者它是 i 的变节起关键作用的赢联合的比例. 这一势力指数避免了构成 Shapley-Shubik 势力指数计算基础的看上去额外的顺序概念.

有趣的是，Shapley-Shubik 指数 p_i^S 可以通过类似与一个(5.37)式的公式计算. 因为 Shapley [1953]证明

$$p_i^S=\sum_S\{\gamma(s)[v(S)-v(S-\{i\})];S\text{ 使得 }i\in S\},\tag{5.38}$$

其中，

$$s=|S|\text{ 和 }\gamma(s)=\frac{(s-1)!(n-s)!}{n!}$$

(参见 2.15 节的练习 15).

Coleman 势力指数的一个变形是 Banzhaf 势力指数(Banzhaf[1965])，其定义如下：

$$P_i^B=\frac{\sum_S[v(S)-v(S-\{i\})]}{\sum_{j=1}^n\sum_S[v(S)-v(S-\{j\})]}.\tag{5.39}$$

这一指数与 Coleman 指数有相同的分子，但是，分母是所有玩家 j 的分子部分的和. 因此，P_i^B 是玩家 i 的关键变节数量除以所有玩家的关键变节数量的总和，或玩家 i 占所有关键变节数量的比例[⊖].

作为一个例子，考虑游戏[51；49，48，3]. 这里，赢联合是{1，2，3}，{1，2}，{1，3}和{2，3}. 玩家 1 的变节对联合{1，2}和{1，3}是关键的，所以我们有 Coleman 指数

$$P_1^C=\frac{2}{4}=\frac{1}{2}.$$

类似地，每一玩家的变节都对两个数的联合是关键的，所以

$$P_2^C=\frac{2}{4}=\frac{1}{2}$$

$$P_3^C=\frac{2}{4}=\frac{1}{2}.$$

注意，在 Coleman 指数中，势力 P_i^C 的总和可能不等于 1. 这是一个相对值. 各 Banzhaf 指数由下式给出：

$$P_1^B=\frac{2}{6}=\frac{1}{3}$$

$$P_2^B=\frac{2}{6}=\frac{1}{3}$$

334

$$P_3^B=\frac{2}{6}=\frac{1}{3}.$$

对于所有三个玩家的势力相同时，上述两个指数与 Shapley-Shubik 指数一致. 不难给出这样的例子：其中这两个指数不同于 Shapley-Shubik 指数(参见练习 2). (当引入若干度量某事件的方法

⊖ 关于 Shapley-Shubik、Banzhaf 和 Coleman 指数的统一框架可以参见 Straffin[1980]. 对于 Shapley-Shubik 指数的文献概览，可以参见 Shapley[1981]. 对于 Banzhaf 和 Coleman 指数的文献概览，可以参见 Dubey 和 Shapley [1979]. 对于这三个指数的应用可以参见 Brams Lucas Straffin[1983]、Brams，Schotter 和 Schwödianer [1979]、Lucas[1981，1983]、Johnston[1995]. 也可参见 2.15.5 节.

而这些指数不同时，我们如何从中选择呢？一个方法是规定若干合理度量应该满足的条件或公理. 于是，我们就能通过检测不同的度量来检查它们是否满足这些公理. Shapley[1953]发现了一组只有 Shapley-Shubik 指数满足的公理；参见 Owen[1995]、Dubey[1975]、Myerson[1997]、Roberts[1976]和 Shapley[1981]. Dubey 和 Shapley[1979]发现了另一组只有 Banzhaf 指数满足的公理；也可参见 Owen[1978a，b]和 Straffin[1980]. Felsenthal 和 Machover[1995]研究了势力指数的公理方法并进行了扩展. 公理方法可能是说服立法委员会或法院使用一种度量而不是其他度量的最合理过程，因为这样立法委员会就可以决定他们是否喜欢特定的一般条件，而不用讨论过程. 顺便提到的是，正是 Banzhaf 指数发现了在法院和一人一票案例中的用途；参见 Lucas[1983].）

我们可以使用生成函数来计算加权多数游戏$[q;\ v_1,\ v_2,\ \cdots,\ v_n]$中的 P_i^C 和 P_i^B 的分子(练习 3 要求读者描述如果确定前者的分母. 如果所有分子是已知的，那么后者的分母的计算是很显然的). 假设玩家 i 的票数为 v_i，如果这一玩家的变节来自于票数是 q 或 $q+1$ 或…，或 $q+v_i-1$ 的联合，那么他的变节就是关键的. 他在这些情况中的变节将导致一个票数为 $q-v_i$，$q-v_i+1$，或…或 $q-1$ 的联合. 假设 $a_k^{(i)}$ 正好是有 k 张票的且不包含玩家 i 的联合的数量. 那么玩家 i 的变节是关键的联合数量由下式给出：

$$a_{q-v_i}^{(i)}+a_{q-v_i+1}^{(i)}+\cdots+a_{q-1}^{(i)}=\sum_{k=q-v_i}^{q-1}a_k^{(i)}. \tag{5.40}$$

假定我们可以计算数 $a_k^{(i)}$，那么在计算 Coleman 或 Banzhaf 指数时上式可以替换成

$$\sum_S[v(S)-v(S-\{i\})].$$

Brams 和 Affuso[1976]指出，使用普通生成函数可以确定 $a_k^{(i)}$. 为了形成一个联合，玩家 j 或者投 0 票或者投 v_j 票. 因此，$a_k^{(i)}$ 的普通生成函数由下式给出：

$$G^{(i)}(x)=(1+x^{v_1})(1+x^{v_2})\cdots(1+x^{v_{i-1}})(1+x^{v_{i+1}})\cdots(1+x^{v_n})=\prod_{j\neq i}(1+x^{v_j}).$$

数 $a_k^{(i)}$ 是上述表达式中 x^k 的系数. 335

以加权多数游戏[4；1，2，4]为例. 我们有

$$G^{(1)}(x)=(1+x^2)(1+x^4)=1+x^2+x^4+x^6$$
$$G^{(2)}(x)=(1+x)(1+x^4)=1+x+x^4+x^5$$
$$G^{(3)}(x)=(1+x)(1+x^2)=1+x+x^2+x^3.$$

因此，例如，$a_4^{(i)}$ 是 $G^{(1)}(x)$中 x^4 的系数，即为 1. 有一个不包含玩家 1 且正好有 4 张选票的联合；即只包含第三个玩家的联合. 使用(5.40)式我们得到

$$\sum_S[v(S)-v(S-\{1\})]=a_{4-1}^{(1)}=a_3^{(1)}=0$$

$$\sum_S[v(S)-v(S-\{2\})]=a_{4-2}^{(2)}+a_{4-2+1}^{(2)}=a_2^{(2)}+a_3^{(2)}=0$$

$$\sum_S[v(S)-v(S-\{3\})]=a_{4-4}^{(3)}+a_{4-4+1}^{(3)}+a_{4-4+2}^{(3)}+a_{4-4+3}^{(3)}$$
$$=a_0^{(3)}+a_1^{(3)}+a_2^{(3)}+a_3^{(3)}=4.$$

上式直接给出

$$P_1^B=\frac{0}{4}=0$$

$$P_2^B=\frac{0}{4}=0$$

$$P_3^B=\frac{4}{4}=1.$$

根据 Banzhaf 指数，玩家 3 有所有的势力．其意义是：任意不包含他的联合都不能赢．Coleman 指数和 Shapley-Shubik 指数产生相同的结果．计算留给读者．

本节练习

1. 对于下列每一个游戏，通过使用生成函数计算分子来计算 Banzhaf 和 Coleman 势力指数．使用这些指数的定义验证你的答案．
   (a)[51; 51, 48, 1]　(b)[51; 49, 47, 4]　(c)[51; 40, 30, 20, 10]
   (d)[20; 1, 10, 10, 10]　(e)[102; 80, 40, 80, 20]
   (f)澳大利亚政府"游戏"(2.15.1 节)：[5; 1, 1, 1, 1, 1, 1, 3]
   (g)1964 年纽约纳苏郡郡议会：[59; 31, 31, 21, 28, 2, 2,]
2. 给出一个游戏的例子，其中：
   (a)Banzhaf 和 Coleman 的势力指数不同．
   (b)Banzhaf 和 Shapley-Shubik 的势力指数不同．
   (c)Coleman 和 Shapley-Shubik 的势力指数不同．
   (d)三个势力指数都不相同．

336

3. 描述如何通过生成函数求解 $\sum_{S} v(S)$．
4. 使用等式(5.38)的公式计算练习 1(a)～(e)中的每一个加权多数游戏的 Shapley-Shubik 势力指数．
5. (a)解释你如何使用生成函数计算 Shapley-Shubik 势力指数．
   (b)把你的结果运用于练习 1 的游戏．

参考文献

BANZHAF, J. F., III, "Weighted Voting Doesn't Work: A Mathematical Analysis," *Rutgers Law Rev., 19* (1965), 317–343.

BERGE, C., *Principles of Combinatorics*, Academic Press, New York, 1971.

BRAMS, S. J., and AFFUSO, P. J., "Power and Size: A Near Paradox," *Theory and Decision, 1* (1976), 68–94.

BRAMS, S. J., LUCAS, W. F., and STRAFFIN, P. D. (eds.), *Political and Related Models*, Modules in Applied Mathematics, Vol. 2, Springer-Verlag, New York, 1983.

BRAMS, S. J., SCHOTTER, A., and SCHWÖDIANER, G. (eds.), *Applied Game Theory*, IHS-Studies No. 1, Physica-Verlag, Würzburg, 1979.

BRESSOUD, D., and WAGON, S., *A Course in Computational Number Theory*, Key College Publishing, Emeryville, CA, 2000.

COHEN, D. I. A., *Basic Techniques of Combinatorial Theory*, Wiley, New York, 1978.

COLEMAN, J. S., "Control of Collectivities and the Power of a Collectivity to Act," in B. Lieberman (ed.), *Social Choice*, Gordon and Breach, New York, 1971, 269–300.

DANIEL, W. W., *Biostatistics: A Foundation for Analysis in the Health Sciences*, 6th ed., Wiley, New York, 1995.

DUBEY, P., "On the Uniqueness of the Shapley Value," *Int. J. Game Theory, 4* (1975), 131–140.

DUBEY, P., and SHAPLEY, L. S., "Mathematical Properties of the Banzhaf Power Index," *Math. Oper. Res., 4* (1979), 99–131.

FELLER, W., *An Introduction to Probability Theory and Its Applications*, 3rd ed., Wiley, New York, 1968.

FELSENTHAL, D. S., and MACHOVER, M., "Postulates and Paradoxes of Relative Voting Power – A Critical Re-Appraisal," *Theory and Decision, 38* (1995), 195–229.

GALLIAN, J. A., *Contemporary Abstract Algebra*, 5th ed., Houghton Mifflin, Boston, 2002.

GARDNER, M., "Mathematical Games," *Scientific American, 238* (1978), 19–32.

GIBBONS, R. D., CLARK, D. C., and FAWCETT, J., "A Statistical Method for Evaluating Suicide Clusters and Implementing Cluster Surveillance," *Am. J. Epidemiol., 132*

(1990), S183–S191.
GOODMAN, A. W., and RATTI, J. S., *Finite Mathematics with Applications*, 4th ed., Prentice Hall, Upper Saddle River, NJ, 1992.
HARARY, F., and PALMER, E. M., *Graphical Enumeration*, Academic Press, New York, 1973.
HARDY, G. H., and WRIGHT, E. M., *An Introduction to the Theory of Numbers*, 5th ed., Oxford University Press, New York, 1980.
JOHNSTON, R. J., "The Conflict over Qualified Majority Voting in the European Union Council of Ministers: An Analysis of the UK Negotiating Stance Using Power Indices," *British J. Pol. Science, 25* (1995), 245–288. 337
KEMENY, J. G., SNELL, J. L., and THOMPSON, G. L., *Introduction to Finite Mathematics*, Prentice Hall, Englewood Cliffs, NJ, 1974.
LANDO, S. K., *Lectures on Generating Functions*, Student Mathematical Library, Vol. 23, American Mathematical Society, Providence, RI, 2003.
LUCAS, W. F., "Applications of Cooperative Games to Equitable Allocation," in W. F. Lucas (ed.), *Game Theory and Its Applications*, Proceedings of Symposia in Applied Mathematics, Vol. 24, American Mathematical Society, Providence, RI, 1981, 19–36.
LUCAS, W. F., " Measuring Power in Weighted Voting Systems," in S. J. Brams, W. F. Lucas, and P. D. Straffin (eds.), *Political and Related Models*, Modules in Applied Mathematics, Vol. 2, Springer-Verlag, New York, 1983, 183–238.
MACMAHON, P. A., *Combinatory Analysis*, Vols. 1 and 2, The University Press, Cambridge, 1915. (Reprinted in one volume by Chelsea, New York, 1960.)
MYERSON, R. B., *Game Theory*, Harvard University Press, Cambridge, MA, 1997.
NIVEN, I., "Formal Power Series," *Amer. Math. Monthly, 76* (1969), 871–889.
OWEN, G., "Characterization of the Banzhaf-Coleman Index," *SIAM J. Applied Math., 35* (1978), 315–327. (a)
OWEN, G., " A Note on the Banzhaf-Coleman Axioms," in P. Ordeshook (ed.), *Game Theory and Political Science*, New York University Press, New York, 1978, 451–461. (b)
OWEN, G., *Game Theory*, 3rd ed., Academic Press, San Diego, CA, 1995.
PARZEN, E., *Modern Probability Theory and Its Applications*, Wiley, New York, 1992.
READ, R. C., "Euler Graphs on Labelled Nodes," *Canad. J. Math., 14* (1962), 482–486.
RIORDAN, J., "Generating Functions," in E. F. Beckenbach (ed.), *Applied Combinatorial Mathematics*, Wiley, New York, 1964, 67–95.
RIORDAN, J., *An Introduction to Combinatorial Analysis*, Princeton University Press, Princeton, NJ, 1980.
ROBERTS, F. S., *Discrete Mathematical Models, with Applications to Social, Biological, and Environmental Problems*, Prentice Hall, Englewood Cliffs, NJ, 1976.
ROSS, S. M., *A First Course in Probability*, 5th ed., Prentice Hall, Upper Saddle River, NJ, 1997.
SHAPLEY, L. S., "A Value for n-Person Games," in H. W. Kuhn and A. W. Tucker (eds.), *Contributions to the Theory of Games*, Vol. 2, Annals of Mathematics Studies No. 28, Princeton University Press, Princeton, NJ, 1953, 307–317.
SHAPLEY, L. S., "Measurement of Power in Political Systems," in W. F. Lucas (ed.), *Game Theory and Its Applications*, Proceedings of Symposia in Applied Mathematics, Vol. 24, American Mathematical Society, Providence, RI, 1981, 69–81.
SRIVASTAVA, H. M., and MANOCHA, H. L., *A Treatise on Generating Functions*, Halsted Press, New York, 1984.
STRAFFIN, P. D., JR., *Topics in the Theory of Voting*, Birkhäuser-Boston, Cambridge, MA, 1980.
TOMESCU, I., *Problems in Combinatorics and Graph Theory*, Wiley, Chichester, UK, 1985.
WILF, H. S., *generatingfunctionology*, Academic Press, Boston, 1994. 338

第6章 递推关系⊖

6.1 一些例子

在5.1节的一开始，我们看到经常要计数依赖于输入或参数k的量a_k. 然后，我们研究了未知值a_0，a_1，a_2，…，a_k，…的序列. 我们将看到如何把这样的序列中的第k项或第$(k+1)$个成员的计算归约成对这一序列的更前面成员的计算. 使用这一方法，我们能够把较大的问题归约成较小的问题或前面已解决的问题. 在3.4节和例5.14中，通过化简定理，我们实际上已经做了很多相同的事情，即将复杂的计算化简成更加简单的或以前已解决的计算. 在知道如何把一个序列的后面的项的计算归约成前面的项的计算后，我们讨论几个寻找未知序列第k项的一般公式的方法. 我们给出的思想和方法有着广泛的应用.

6.1.1 一些简单的递推关系

【例6.1 小麦数量】 根据Gamow[1954]，下面是印度国王Shirham的故事. 这位国王想要奖赏他的大臣Sissa Ben Dahir，因为他发明了国际象棋游戏. 国王问大臣有何要求，这位大臣提出一个谨慎的要求：在一个棋盘的第一个格子里放一粒小麦，第二个格子里放两粒小麦，第三个格子里放四粒小麦，第四个格子里放八粒小麦，以此类推，直到所有格子放完为止. 国王对大臣的这一谨慎的要求很满意，因此立即就允诺了. 这位国王做了一件很聪明的事情吗？为了回答这一问题，设s_k是前k个格子所需的小麦数量，而t_k是第k个格子所需的小麦数量. 我们有

339

$$t_{k+1} = 2t_k. \tag{6.1}$$

等式(6.1)是*递推关系*(recurrence relation)的一个例子，这样的关系是把一个数序列的后项值归约成前项值的公式. 让我们看一下如何使用这一递推公式得到t_k的一般表达式. 我们知道$t_1=1$. 这是给出的，称为*初始条件*(initial condition). 我们知道

$$\begin{aligned} t_2 &= 2t_1 \\ t_3 &= 2t_2 = 2^2 t_1 \\ t_4 &= 2t_3 = 2^2 t_2 = 2^3 t_1, \end{aligned}$$

一般地，有

$$t_k = 2t_{k-1} = \cdots = 2^{k-1} t_1.$$

使用上面的初始条件，对于所有的k，我们有

$$t_k = 2^{k-1}. \tag{6.2}$$

我们已经通过这一递推关系的*迭代*(iteration)解出了递推关系式(6.1). 注意，像(6.1)式这样的递推关系式一般有很多*解*(solution)，即有很多满足这一关系的序列. 然而，一旦给出足够多的初始条件，那么将存在唯一解. 这里，序列1，2，4，8，…是上面给定的初始条件下的唯一解. 然而，如果忽视这一初始条件，那么上面的序列的任意倍数都是一个解，如3，6，12，24，…或5，10，20，40，…等.

我们感兴趣的实际上是s_k. 我们有

⊖ 如果跳过了第5章，那么6.3节和6.4节也应该跳过. 第5章和第6章是除需一定的数学知识外，还需要微积分知识的唯一章节. 除了6.3节和6.4节之外，第6章所使用的微积分仅仅是关于无穷序列的初步知识；即使在6.3节和6.4节，我们也只是在几个应用中使用了极限的概念，这些内容可以跳过.

$$s_{k+1}=s_k+t_{k+1},\tag{6.3}$$

这是递推关系的另外一种形式，它把 s 的后项值与 s 的前项值和已计算出来的 t 的值联系起来. 使用(6.2)式我们可以把(6.3)式化简成一个仅与 s_k 有关的递推关系. 这给出下面的表达式:

$$s_{k+1}=s_k+2^k.\tag{6.4}$$

再一次使用迭代对于所有的 k 求解 s_k 的递推关系式(6.4). 我们有

$$s_2=s_1+2$$
$$s_3=s_2+2^2=s_1+2+2^2,$$

一般地，有

$$s_k=s_{k-1}+2^{k-1}=\cdots=s_1+2+2^2+\cdots+2^{k-1}.$$

因为我们有初始条件 $s_1=1$，所以得到

$$s_k=1+2+2^2+\cdots+2^{k-1}.$$

如果我们使用下面著名的恒等式，那么上面的表达式可以得到化简，这个著名的恒等式我们已在第5章中遇到过:

$$1+x+x^2+\cdots+x^p=\frac{1-x^{p+1}}{1-x}.$$

340

对 $x=2$ 和 $p=k-1$ 使用这一恒等式，有

$$s_k=\frac{1-2^k}{1-2}=2^k-1.$$

现在棋盘上有64个格子. 因此，那位大臣要求的小麦数量是 $2^{64}-1$，这个数字是

18 446 744 073 709 551 615,

这实际上是一个非常大的数字[⊖]. ■

【例6.2 计算复杂度】 在计算机科学中递推关系的一个主要应用是计算一个算法在输入大小为 n 时的复杂度 $f(n)$(参见2.4节). 通常，复杂度 $f(n+1)$ 的计算被归约成关于复杂度 $f(n)$, $f(n-1)$ 等的知识. 作为一个显然的例子，让我们考虑下面求一个序列或数组的前 n 项和的算法.

算法6.1：求一个序列或数组的前 n 项和

输入：一个序列或数组和一个数字 n.

输出：和 $A(1)+A(2)+\cdots+A(n)$.

步骤1. 令 $i=1$.

步骤2. 令 $T=A(1)$.

步骤3. 如果 $i=n$，停止并输出 T. 否则，设 $i=i+1$，并进入到步骤4.

步骤4. 设 $T=T+A(i)$，并返回到步骤3.

如果 $f(n)$ 是对 A 的前 n 项求和中所执行的加法的数量，那么我们有递推关系

$$f(n)=f(n-1)+1.\tag{6.5}$$

同样，我们也有初始条件 $f(1)=0$. 因此，通过迭代，我们有

$$f(n)=f(n-1)+1=f(n-2)+1+1=\cdots=f(1)+n-1=n-1.$$ ■

【例6.3 单利与复利】 假设把总额为 S_0 的钱按着每个利息周期(比如说每一年)以利率 r 即

⊖ 本章的所有序列都可以在整数序列在线百科全书中找到(Sloane[2003]). 这是含盖超过90 000个序列的数据库. 对于每一个序列，数据库给出这一序列的开始若干项、它的名字或描述、参考文献、公式，等等.

341 按百分之 100r 存入银行. 如果利息是单(simple)利，那么每一个利息周期之后，初始存款 S_0 的 100r%被加入到账户. 如果 S_k 是 k 个周期后的存款总数，那么我们有下面的递推关系：

$$S_{k+1} = S_k + rS_0. \tag{6.6}$$

通过迭代，我们有

$$S_k = S_{k-1} + rS_0 = S_{k-2} + rS_0 + rS_0 = \cdots = S_0 + krS_0,$$

所以有

$$S_k = S_0(1+kr).$$

如果利息是每一周期被累加的复利(compounded)，那么每一个周期之后，我们得到这一周期开始时的存款额的百分之 100r 的利息，即我们有下面的递推关系：

$$S_{k+1} = S_k + rS_k,$$

或

$$S_{k+1} = (1+r)S_k. \tag{6.7}$$

通过迭代我们有

$$S_k = (1+r)^k S_0.$$ ■

【例 6.4 合法代码字】 数字表{0，1，2，3}上的代码字被认为是合法的(legitimate)，当且仅当它们有偶数个 0. 存在多少长度为 k 的合法代码字呢？设 a_k 是长度为 k 的合法代码字的数量. 我们推导一个关于 a_k 的递推关系(注意，使用第 5 章的生成函数的方法可以计算 a_k). 通过观察可知，4^k-a_k 是长度为 k 的不合法代码字的数量，即有奇数个 0 的 k 个数字的代码字. 考虑一个长度为(k+1)的合法的代码字. 它开始于是 1，2 或 3，或开始于 0. 对于前者，后 k 个数字形成一个长度为 k 的合法代码字，而对后者，后 k 个数字却形成一个长度为 k 的不合法代码字. 因此，通过第 2 章的乘法和加法规则，有

$$a_{k+1} = 3a_k + 1(4^k - a_k),$$

即

$$a_{k+1} = 2a_k + 4^k. \tag{6.8}$$

我们有初始条件 $a_1=3$. 求解递推关系(6.8)式的一个方法是通过迭代方法. 这类似于递推关系(6.4)式的求解过程，我们将此工作留给读者. 求解递推关系(6.8)式的另一个方法将在 6.3 节给出. 现在，我们计算一些 a_k 的值. 注意，因为 $a_1=3$，这一递推关系给出我们下面的结果：

$$a_2 = 2a_1 + 4^1 = 2(3) + 4 = 10$$

和

342

$$a_3 = 2a_2 + 4^2 = 2(10) + 16 = 36.$$

读者也许希望通过写出长度为 2 和 3 的合法代码字来核实这些数字. 注意我们是如何利用 a_k 前面的值来得到后面的值的. 使用一个递推关系计算一些未知数无需一个明确的解. ■

【例 6.5 信息的传输时间】 设想我们在一个信道上只使用两个信号 a 和 b 来传输信息. 一个代码字是由字母表{a，b}形成的任意序列. 现在，假设传输信号 a 需要花费 1 个时间单位，而传输信号 b 需要花费 2 个时间单位. 设 N_t 是恰好在 t 个时间单位内可以传输的代码字的可能数量. N_t 是多少呢？为了回答这一问题，考虑一个在 t 个时间单位内可以传输的代码字. 它或开始于 a 或开始于 b. 如果它开始于 a，那么剩余部分是可以在 $t-1$ 个单位时间内传输的任意代码字. 如果它开始于 b，那么其剩余部分是在 $t-2$ 个时间单位内传输的任意代码字. 因此，根据加法规则，对于$t\geq 2$，

$$N_t = N_{t-1} + N_{t-2}. \tag{6.9}$$

这是第一个给定值依赖于多个前面值的递推关系的例子. 对于这一递推关系，因为第 t 项依赖于其前面两项的值，所以我们需要两个初始条件 N_1 和 N_2. 显然，$N_1=1$，$N_2=2$，后者是因为在 2 个时间单位内 aa 和 b 是两个唯一可传输的序列. 当我们开发若干求解递推关系的一般工具之后，我们将在 6.2 节中求解递推关系(6.9)式. Shannon[1956]如下定义传输信道的容量(capacity)C.

$$C=\lim_{t\to\infty}\frac{\log_2 N_t}{t}.$$

上面的式子是信道传输信息的容量的一种度量. 我们将在 6.2.2 节中讨论这一定义. ■

【例 6.6 排列的互换平均值】 给定集合$\{1, 2, 3, \cdots, n\}$的一个排列 π，Jerrum 公式(2.17)计算把恒等排列转换成 π 所需的互换数量. 如果 π 是随机选取的，那么所需的期望互换数量是多少？或者，换一种说法，把恒等排列转换成$\{1, 2, \cdots, n\}$上的一个排列 π 所需的互换平均值是多少？

假设$\{1, 2, \cdots, n\}$的每一个排列都是均等的. 设 b_n 等于把恒等排列转换成$\{1, 2, \cdots, n\}$上的一个排列所需的互换平均值. 例如，如果 $n=2$，有$\{1, 2\}$的两个排列，即为 12 或 21. 把恒等排列转换成 12 不需要互换. 而把它转换成 21 需要一次互换. 所需的互换平均值是 $b_2=(0+1)/2=1/2$. 为了确定 b_{n+1}，考虑$\{1, 2, \cdots n\}$上的任意排列 $\pi=\pi_1\pi_2\cdots\pi_n$，并对于 $i=1, 2, \cdots, n+1$，设 π^i 是 $\pi_1\pi_2\cdots\pi_{i-1}(n+1)\pi_i\cdots\pi_n$. [如果 $i=n+1$，$\pi^i=\pi_1\pi_2\cdots\pi_{n(n+1)}$.]设 $\alpha(\pi)$是把恒等排列转换成 π 所需的互换数，且 $\alpha(\pi^i)$是把恒等排列转换成 π^i 所需的互换数. 注意在 π^i 中，$n+1$ 比其右侧的任意一个数字都大. 因此，根据 Jerrum 公式(2.17)，有：$\alpha(\pi^i)=\alpha(\pi)+[n-(i-1)]$，所以根据算术级数求和的标准公式， [343]

$$\begin{aligned}\sum_{i=1}^{n+1} a(\pi^i) &=[(a(\pi)+n)+(a(\pi)+(n-1))+\cdots+(a(\pi)+0)]\\ &=[(n+1)a(\pi)+n+(n-1)+\cdots+0]\\ &=[(n+1)a(\pi)+n(n+1)/2],\end{aligned}$$

于是有

$$\begin{aligned}b_{n+1} &=\frac{\sum_{\pi}\sum_{i=1}^{n+1}a(\pi^i)}{(n+1)!}=\frac{\sum_{\pi}[(n+1)a(\pi)+n(n+1)/2]}{(n+1)!}\\ &=\frac{\sum_{\pi}a(\pi)}{n!}+\frac{n!(n(n+1)/2)}{(n+1)!}\\ &=b_n+\frac{n}{2}.\end{aligned}$$

为了求解这一递推关系，我们注意到 $b_1=0$，$b_2=1/2$，且

$$b_3=b_2+\frac{2}{2}=\frac{1}{2}+\frac{2}{2},$$

$$b_4=b_3+\frac{3}{2}=\frac{1}{2}+\frac{2}{2}+\frac{3}{2},$$

$$b_5=b_4+\frac{4}{2}=\frac{1}{2}+\frac{2}{2}+\frac{3}{2}+\frac{4}{2}.$$

一般地，再一次使用算术级数的标准求和公式有

$$b_{n+1}=b_n+\frac{n}{2}=\cdots=\frac{1}{2}+\frac{2}{2}+\frac{3}{2}+\cdots+\frac{n}{2}=\frac{1}{2}(1+2+\cdots+n)=\frac{n(n+1)}{4}.$$ ■

【例 6.7 平面中的区域】 一条直线把平面分割成两个区域(参见图 6.1). 两条相交直线把平面分割成四个区域(参见图 6.1). 假设我们有 n 条处于“一般位置”的直线；即没有两条直线平行，也没有三条直线相交于同一点. 这些直线可以把平面分割成多少个区域呢？为了回答这一问题，设 $f(n)$ 是所求的区域数. 我们已看到 $f(1)=2, f(2)=4$. 图 6.1 也已给出 $f(3)=7$. 为了确定 $f(n)$. 我们将推导一个递推关系.

344

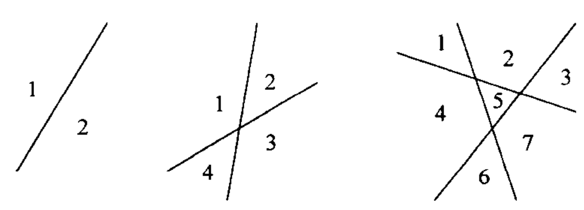


图 6.1 把平面分割成区域的直线

考虑图 6.2 所示的直线 l_1. 画出第二条直线 l_2. 直线 l_1 把直线 l_2 分割成两个部分，而且每一个部分把已存在的区域分成两个区域. 因此

$$f(2) = f(1) + 2.$$

类似地，如果我们增加第三条直线 l_3，l_1 和 l_2 把这一直线分成三个部分，且每一部分又把已存在的区域分成两部分(参见图 6.3). 因此

$$f(3) = f(2) + 3.$$

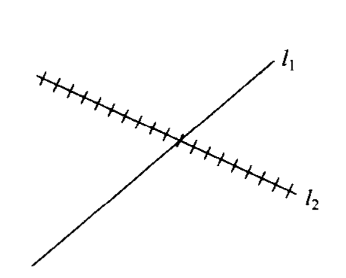


图 6.2 直线 l_1 把直线 l_2 分成两个部分

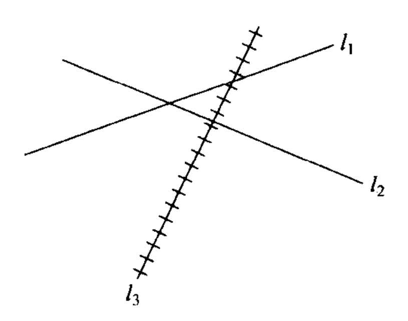


图 6.3 直线 l_1 和直线 l_2 把直线 l_3 分成三个部分

一般地，假设把直线 l_{n+1} 加入到已存在的直线 l_1，l_2，$\cdots$ l_n 中. 已存在的直线把 l_{n+1} 分成 $n+1$ 个部分，每一个部分把已存在的区域分成两个部分(参见图 6.4). 因此，

$$f(n+1) = f(n) + (n+1). \tag{6.10}$$

图 6.4 直线 l_1，l_2，$\cdots l_n$ 把直线 l_{n+1} 分割成 $n+1$ 个部分

等式(6.10)给出我们将用于求解 $f(n)$ 的递推关系. 其初始条件是 $f(1)$ 的值，即 2. 为了求解递推关系(6.10)式，注意到有下面的关系式：

345

$$f(2) = f(1) + 2,$$

$$f(3) = f(2) + 3 = f(1) + 2 + 3,$$

$$f(4) = f(3) + 4 = f(2) + 3 + 4 = f(1) + 2 + 3 + 4,$$

一般地，有

$$f(n) = f(n-1) + n = \cdots = f(1) + 2 + 3 + \cdots + n.$$

因为 $f(1)=2$，再一次使用算术级数标准求和公式，我们有

$$\begin{aligned} f(n) &= 2+2+3+4+\cdots+n \\ &= 1+(1+2+3+4+\cdots+n) \\ &= 1+\frac{n(n+1)}{2}, \end{aligned}$$

例如，我们有

$$f(4) = 1+\frac{4\cdot 5}{2} = 11.$$

■

6.1.2 斐波那契数及其应用

1202年，以斐波那契著称的比萨的列昂纳多在他的著作《算盘书》(Liber Abaci)中提出了下面的问题．假设我们研究兔子的增值饲养．我们先从一对(性别相反的)成年兔开始．假设每一对成年兔每一个月可以产下一对性别相反的幼兔．新出生的幼兔在两个月内长大为成年兔，在这段时间内它们也产下它们自己的第一对幼兔．假设兔子没有死亡的．设 F_k 计数第 k 个月开始时所存在的兔子对面数量．表6.1列出对应若干 k 值的成年兔对的数量，一个月大的幼兔对数量，新生幼兔对的数量和兔对的总数量．例如，在第二个月开始时，有一对新生幼兔．在第三个月开始时，前一个月的新生幼兔变成一个月大的兔子，而且又有一对新生幼兔出生．在第四个月开始时，一个月大的兔子变成成年兔，而且出生新生幼兔，所以到目前为止有两对成年兔和两对新生兔．第三个月的一对新生兔已经是一个月大的一对兔子等等．

346

表6.1 兔子繁殖

月份 k	第 k 个月开始时成年兔对的数量	第 k 个月开始时一个月大的兔对的数量	第 k 个月开始时新生兔对的数量	第 k 个月开始时兔对的总数量
1	1	0	0	1
2	1	0	1	2
3	1	1	1	3
4	2	1	2	5
5	3	2	3	8
6	5	3	5	13

我们推导 F_k 的递推关系．注意在第 k 个月开始时兔对的数量等于第 $(k-1)$ 个月开始时兔对的数量加上第 k 个月开始时新出生兔对的数量．但是后一个数字与第 k 个月开始时成年兔对的数量相等，而这一成年兔对的数量又等于第 $(k-2)$ 个月开始时所有兔对的数量(要花正好两个月时间才能变成成年兔)．因此，对于 $k\geqslant 3$，我们有

$$F_k = F_{k-1} + F_{k-2}. \tag{6.11}$$

注意，如果我们定义 F_0 等于1，那么对 $k\geqslant 2$，(6.11)式成立．观察递推关系(6.11)式和(6.9)式的相似之处．我们将在6.2.2节返回这一问题．使用递推关系(6.11)式计算 F_k 的几个值．我们已经知道

$$F_0 = F_1 = 1.$$

因此

$$\begin{aligned} F_2 &= F_1 + F_0 = 2, \\ F_3 &= F_2 + F_1 = 3, \\ F_4 &= F_3 + F_2 = 5, \end{aligned}$$

$$F_5 = F_4 + F_3 = 8,$$
$$F_6 = F_5 + F_4 = 13,$$
$$F_7 = F_6 + F_5 = 21,$$
$$F_8 = F_7 + F_6 = 34.$$

在 6.2.2 节，我们将使用递推关系(6.11)式得到数 F_k 的明确公式. 数 F_0，F_1，$F_2\cdots$的序列被称为*斐波那契序列*(Fibonacci sequence)，各 F_k 称为*斐波那契数*(Fibonacci number). 这些数有很多引人注目的性质且在很多地方出现. 这里，我们将给出其中的一些性质及其应用.

347

序列(F_k)在 k 时刻时的*生长率*(grouth rate)定义如下：

$$G_k = \frac{F_k}{F_{k-1}}.$$

于是我们有

$$G_1 = \frac{1}{1} = 1, G_2 = \frac{2}{1} = 2, G_3 = \frac{3}{2} = 1.5, G_4 = \frac{5}{3} = 1.67, G_5 = \frac{8}{5} = 1.60,$$

$$G_6 = \frac{13}{8} = 1.625, G_7 = \frac{21}{13} = 1.615, G_8 = \frac{34}{21} = 1.619, \cdots.$$

数 G_k 似乎收敛于 1.60 与 1.62 之间的一个极限. 事实上，这个极限的精确值是

$$\tau = \frac{1}{2}(1+\sqrt{5}) = 1.6180\,34\cdots.$$

这个数 r 将出现在 F_k 的一般推导公式中[这个数称为*黄金比例*(golden ratio)或*神圣的比例*(divine proportion)]. 这是一个有如下性质的数：如果你把直线 AB 在 C 点分开使得 $\tau=AB/AC$，那么

$$\frac{AB}{AC} = \frac{AC}{CB}.$$

边的比值为 $\tau:1$ 的矩形称为*黄金矩形*(golden rectangle). 15 世纪艺术家 Piero della Francesca 在他的书籍《De Divina Proportione》(神圣的比例)中通篇描述了 τ 以及黄金矩形在艺术中的应用，特别是在达芬奇的作品中的应用. 尽管黄金比例被大量用于艺术、建筑以及美学，但是已经证明大多黄金比例的断言是“假的或严重误导”(Markowsky[1992]).

斐波那契数在数值分析中有着重要的应用，特别是对寻找函数 $f(x)$在区间(a，b)内的最大值有重要的应用. 使用计算机的最大值*斐波那契搜索*(Fibonacci search)就是利用斐波那契数来确定在什么地方评估函数以便最大值位置的评估越来越好. 当 f 是凹函数时，在最小化寻找到达一个理想精度最大值所需的函数评估数量的意义下，这一搜索法可能是已知的最好的搜索过程. 参见 Adby 和 Dempster[1974]、Hollingdale[1978]或 Kiefer[1953].

令人好奇的是自然界中频繁出现斐波那契数. 研究树干周围叶子的分布，松果上的鳞片硬壳等的植物学领域称为*叶序*(phyllotaxis). 通常，出现在给定树干或分支上的叶子是指向不同的方向的. 第二片叶子与第一片叶子成某个角度偏转，第三片叶子与第二片叶子成相同的角度偏转，以此类推，直到某个叶子伸展的方向与第一个叶子相同. 例如，如果这一偏转角度是 30 度，那么第十二片叶子是第一个与第一片叶子的伸展方向相同的叶子，因为 12×30=360 度. 如果这一角度是 144 度，那么第五片叶子是第一个与第一片叶子的伸展方向相同的叶子，因为 5×144=720 度. 两个完整的 360 度偏转使得一片叶子的面朝向与第一片叶子相同. 一般地，设 n 是偏转到与第一片叶子相同的伸展方向之前的叶子的数量，设 m 是再次与这片叶子相遇之前所需要的完整的 360 度的圈数. 表 6.2 给出各种植物的 n 和 m 值. 最常见的是 n 和 m 都取值为某个斐波那契数，这是植物学中值得注意的实验事实. 对此事实没有恰当的理论解释.

348

表 6.2 各种植物的 n 和 m 的值

植物	旋转角度	n	m
榆树	180°	2	1
西楠桦	120°	3	1
玫瑰	144°	5	2
卷心菜	135°	8	3

资料来源：Schips[1922]、Batschelet[1971].

Coxeter[1969]指出斐波那契数还出现在对冷杉果子上的鳞片，菠萝上的螺层等的研究中. 这些螺层(蜂房)是以非常明显的对角行排列的. 这些螺层能够以下面的方法指定整数：每一个螺层的对角行形成分差(两个相邻数间的差)为斐波那契数的算术序列. 如图 6.5 所示. 例如，对角行

$$9,22,35,48,61,74,$$

这一序列的分差为斐波那契数 13. 类似地，对角行

$$11,19,27,35,43,51,59$$

有分差 8. 向日葵花盘、冷杉果子的鳞片等也有类似的结论(参见 Adler [1977] 或 Fowler, Prusinkiewicz 和 Battjes [1992]). 同样，对为什么斐波那契数出现于叶序的这一现象也没有恰当的理论解释.

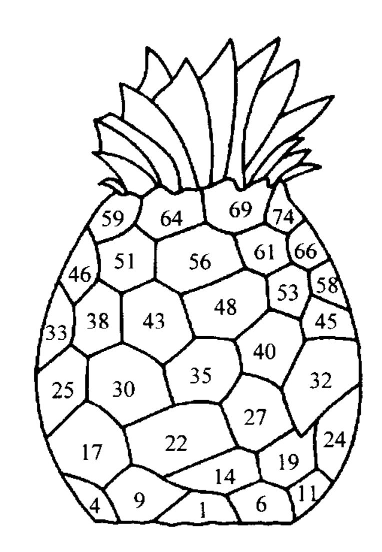


图 6.5 斐波那契数和菠萝(取自 Coxeter[1969]. 经 John Wiley & Sons, Inc 许可)

【例 6.8 "热手效应"(再探例 2.32)】 回想蓝球中所谓的"热手效应"现象. 热手效应假设一旦一名队员投中了球，那么他投中下一个球的几率高于平均几率. 在例 2.32 中，我们考虑了所有投中连续发生的概率. 现在，我们将考虑没有连续投中的方法数量.

如果一名队员投了 n 个球，则有 2^n 个可能的结果；每一次投射或者是投中或者是没有投中㊀. 设 B_n 表示不包含连续投中的 X 和 O 的序列的子集，即没有两个连续 X 的序列的子集. 为了确定 $b_n=|B_n|$，我们把第 i 次投中记为 c_i 并设 $C_n=\{c_1, c_2, \cdots, c_n\}$. 使用另外一种方法来陈述，我们的问题是：不包含 c_i，c_{i+1}，$i=1, 2, \cdots, n-1$ 的 C_n 的子集有多少？设 B 是这样的一个子集. B 或者包含 c_n 或者不包含它. 如果 B 包含 c_n，那么 $B-\{c_n\}$是 C_{n-2}的一个子集，因为 c_{n-1}一定不在 B 中. 如果 B 不包含 c_n，那么 $B-\{c_n\}=B$ 是 C_{n-1}的一个子集. 因此，对于 $n\geqslant 2$，我们有 [349]

$$b_n = b_{n-1} + b_{n-2}. \tag{6.12}$$

因为对 $n=1$，$b_1=2$；一次投射或者投中或者失利. 如果 $n=2$，那么或者没有投中的，或者第一次投球投中而第二次失利，或者反之. 因此，$b_3=3$. 使用这些初始条件，我们看到递推关系(6.12)式的求解与斐波那契数关系密切. 特别地，

$$b_n = F_{n+1}, \quad n \geqslant 1.$$ ■

因此，如果尝试投球 10 次(如例 2.32 那样)，在 $2^{10}=1024$ 个可能结果中只有 $b_{10}=F_{11}=144$ 个不包含连续投中.

6.1.3 错位排列

【例 6.9 衣帽间问题】 想象 n 位绅士出席一次派对，并寄存他们的帽子. 这位收管员喝多了，所以随便地还回了帽子. 所有绅士都没得到他们自己帽子的概率是什么？这一概率与绅士的 [350]

㊀ 这并不表明所有 2^n 个排序都是均等的. 投中球和没有投中球的机会很少是均等的.

数量有什么依赖关系？我们将通过研究错位排列的概念来回答这些问题. 对于这一问题的来源、历史以及另一种阐述，参见 Takacs[1980]. ■

设 n 个对象被标上标签 1，2，…，n. 对于所有的 i，对象 i 都不在第 i 个位置上的排列称为错位排列(derangement). 例如，如果 $n=3$，那么 231 就是一个错位排列，但是 213 不是，因为 3 在第三个位置上. 设 D_n 是 n 个对象的错位排列的数量.

错位排列出现在一种纸牌游戏(rencontres)中：把一副纸牌在桌子上排成一行，而另一副纸牌则在第一副纸牌的每张牌上随机摆放一张纸牌. 匹配的纸牌数量决定点数. 1708 年，Frenchman P. R. Montmort 提出了不匹配发生的概率计算问题，并称其为“le probleme desrencontres”(匹配问题)，recontre 在法语中意思是“匹配”. 当然，这是计算 D_{52} 的问题. 计算匹配纸牌数量的问题将在 7.2 节中讨论. 在那里，我们还讨论在超能力实验中对猜测能力分析的应用. 第一副纸牌对应于所选择，或所取样的事件的一个未知顺序，而第二副纸牌则对应于一个超能者所预测的顺序. 在测试超能力推断时，我们也许想要计算得到正确匹配的概率. 至少得到一个正确匹配的概率是 1 减去没有得到正确匹配的概率，即 1 减去得到一个错位排列的概率.

显然，$D_1=0$：不存在元素不在其正确位置上的一个元素的排列. $D_2=1$：唯一的错位排列是 21. 我们推导关于 D_n 的递推关系. 假设有 $n+1$ 个元素 1，2，…，$n+1$. 这 $n+1$ 个元素的一个错位排列要涉及第一个元素的选择，然后是其余 n 个元素的一个排列. 第一个元素可以是 n 个不同的元素中的任意一个：2，3，…，$n+1$. 假设 k 被放到第一个位置上. 那么 1 或者出现在第 k 个位置上或者不出现在该位置上. 如果 1 出现在第 k 个位置上，那么我们有余下的元素

$$2,3,\cdots,k-1,k+1,\cdots,n+1,$$

而且我们希望排列这余下的元素，使得这些元素中没有一个出现其正确的位置上[参见图 6.6a]. 有 D_{n-1} 种方法可以做到这一点，因为有 $n-1$ 个元素. 下面假设 1 不出现在第 k 个位置上. 我们首先把 1 放在第 k 个位置上[如图 6.6b 所示]，然后错位排列第二个位置到第$(n+1)$个位置上的元素. 有 D_n 个这样的错位排列. 总之，元素 k 出现在第一个位置上我们有 n 种选择. 对于其中的每一种选择，我们或者选择 1 在第 k 个位置上的一个排列，这可以有 D_{n-1} 种方法实现，或者选择 1 不在第 k 个位置上的一个排列，这可以有 D_n 种方法实现. 根据乘法规则和加法规则，

$$D_{n+1}=n(D_{n-1}+D_n). \tag{6.13}$$

[351] 等式(6.13)是对 $n\geqslant 2$ 的所有 n 成立的递推关系. 如果我们把初始条件 $D_1=0$，$D_2=2$ 加进去，这一递推关系变成

$$D_n=n!\left[1-\frac{1}{1!}+\frac{1}{2!}-\frac{1}{3!}+\cdots+(-1)^n\frac{1}{n!}\right], \tag{6.14}$$

或者(对于 $n\geqslant 2$)有

$$D_n=n!\left[\frac{1}{2!}-\frac{1}{3!}+\cdots+(-1)^n\frac{1}{n!}\right]. \tag{6.15}$$

我们将在 6.3.2 节中看到是如何得到这些公式的.

	k				1					
a)	•	…			•	…				
	第一				第 k					
	k	2	3		$k-1$	1	$k+1$		$n+1$
b)	•	•	•	…	•	•	•	…	•
	第一	第二	第三		第$(k-1)$	第 k	第$(k+1)$		第$(n+1)$

图 6.6 关于 k 在第一个位置，1 在第 k 个位置上及其他元素的错位排列，
a) 以任意顺序，b) 以正确位置

现在我们运用这些公式解决例6.9的衣帽间问题. 如果有n位绅士，设P_n是没有绅士拿到自己的帽子的概率. 那么(对于$n\geqslant 2$)有

$$P_n=\frac{\text{所有绅士都没有拿到自己的帽子的排列数量}}{\text{总排列数量}}=\frac{D_n}{n!}$$

$$=\left[\frac{1}{2!}-\frac{1}{3!}+\cdots+(-1)^n\frac{1}{n!}\right].$$

表6.3给出了对应于若干个n的P_n值. 注意P_n似乎很快收敛. 事实上，我们可以精确地计算出P_n收敛于什么. 回想一下：

$$e^x=1+x+\frac{1}{2!}x^2+\frac{1}{3!}x^3+\cdots.$$

[参见(5.3)式]. 因此

$$e^{-1}=\frac{1}{2!}-\frac{1}{3!}+\cdots,$$

所以P_n收敛于$e^{-1}=0.367\,879\,441\cdots$. 这一级数的收敛很快，使得$P_7$和$P_8$在第五位小数上就已经不同. 绅士们没有得到自己帽子的概率则很快就变得基本上与绅士的数量无关. 352

表6.3 P_n的值

n	2	3	4	5	6	7	8
p_n	0.500 000	0.333 333	0.375 000	0.366 667	0.368 056	0.367 858	0.367 883

【例6.10 拉丁矩形】 在第1章，我们讨论了拉丁方及其对实验设计的应用，在5.1节的练习15中，我们介绍了拉丁矩形和逐行构建拉丁方的思想. 设$L(r, n)$是用项1，2，…n构建的$r\times n$拉丁矩形的数量(回想一下，这样的矩阵是一个各项为1，2，…，n的$r\times n$数组，使得在任意行或任意列中都没有两个项是相同的). 设$K(r, n)$是有项1，2，…，n且第一行是标准位置(standard position)123…n的$r\times n$拉丁矩形的数量. 那么

$$L(r,n)=n!K(r,n), \tag{6.16}$$

我们可以通过寻找一个第一行是标准位置的拉丁矩形，然后改变第一行的排列，并对其余各行的元素进行相同的序列改变，就可以得到任意一个拉丁矩形.

我们也许想要对每一个r和n，计算$L(r, n)$和$K(r, n)$. 通过(6.16)式，这些问题是等价的. 在例1.1中，我们要求的是$L(5, 5)$(这个数是61，280；参见Ryser[1963]). $K(2, n)$很容易计算. 它就是D_n，即n个元素的错位排列数量. 因为我们通过错位排列第一行的元素，得到第一行在标准位置上的拉丁矩形的第二行.

基于某些矩阵函数，Fu[1992]给出了两个$L(r, n)$的公式(基于MacMahon[1898]的想法，Shao和Wei[1992]给出了$L(r, n)$的精确公式). ■

【例6.11 DNA序列对位】 正如在2.17节所提示的那样，突变是发生进化的一个关键过程. 给定两个不同物种的DNA序列，我们有时候尝试着要弄明白它们的关系有多近，特别是要寻找在两者都出现的模式. 我们通常通过将这两个DNA序列对位来完成这一工作：一个序列在另外一个序列的下面，使得两个序列的(几乎)相同子序列对齐. 例如，考虑序列AATAATGAC和GAGTAATCGGAT(注意它们的长度不同). 一种对位如下所示：

```
      A A T A A T G A C
    G A G T A A T C G G A T        (6.17)
```

这里，我们发现一个共同的子序列TAAT(在实际的对位应用中，我们通常允许字母的插入和删

除，但是在这里我们忽略这一点）．寻找好的序列对位，即有长的共同子序列或模式的对位，已为生物学带来了辉煌的前景．例如，人们已经发现促进身体成长的血小板衍生因子的序列与一种引发癌症的基因 v-sis 的序列有 87%相同．这导致我们发现 v-sis 具有促进生长的功能．事实上，正如 Gusfield[1997]指出的那样，在 DNA 序列中（以及，更一般的，在诸如 RNA 或氨基酸序列等其他生物分子序列中）"高序列相似性常常意味着重大的功能相似性或结构相似性"．

353

度量两个序列对位好坏的一个方法是计数它们不匹配的位置数量．这个数量称为错配距离（mismatch distance）．例如，(6.17)式的对位中，如果我们忽视顶端序列没有项的位置，那么错配距离为 4，如果我们计数一个序列没有项的位置，那么此对位的错配距离是 7．对于更多分子生物学中序列对位的内容可以参见 Apostolico 和 Giancarlo[1999]、Gusfield[1997]、Myers[1995]、Setubal 和 Meidanis[1997]、Waterman[1995]．

在此，我们考虑序列对位问题的一个简化版本．假设给我们一个长度为 n 的 DNA 序列，而且它通过随机排列元素进化成另一个这样的序列．两个序列的错配距离是 n 的概率是多少呢？如果第一个序列是 AGCT，新序列是第一个序列的错位排列的概率是 $D_4/4!=3/8$．然而，如果原来的序列是 AGCC，其中有重复的项，问题就不同了．首先，对于这些基，现在仅有 $4!/2!=12$ 种可能的序列，而不是 $4!$ 种（为什么）．其次，AGCC 的所有错位排列中只有 CCAG 和 CCGA 这两个排列中没有元素与 AGCC 的位置相同．因此，期望的概率是 $2/4!/2!=1/6$，而不是 $D_4/4!=3/8$．

■

6.1.4　涉及多个序列的递推关系

为了扩展例 6.4，让我们定义数字表{0，1，2，3}上的一个代码字是合法的（legitmate）当且仅当它有偶数个 0 且有偶数个 3．设 a_k 是长度为 k 的合法代码字的数量．我们如何确定 a_k？为了回答这个问题，可以证明，与例 6.4 的方法类似，考虑长度为 k 的代码字的其他概率是很有用的．对于数字表{0，1，2，3}，设 b_k 是长度为 k 且有偶数个 0 和奇数个 3 的代码字的数量，c_k 是有奇数个 0 和偶数个 3 的代码字的数量，d_k 是有奇数个 0 和奇数个 3 的代码字的数量．注意

$$d_k = 4^k - a_k - b_k - c_k. \tag{6.18}$$

观察发现，通过在一个长度为 k 的合法代码字前面添加一个 1 或一个 2，或在长度为 k 且有偶数个 0 和奇数个 3 的代码字之前添加一个 3，或在长度为 k 且有奇数个和 0 偶数个 3 的代码字之前添加一个 0，我们就可以得到一个长度为 $k+1$ 的合法代码字．因此

$$a_{k+1} = 2a_k + b_k + c_k. \tag{6.19}$$

类似地，有

354

$$b_{k+1} = a_k + 2b_k + d_k, \tag{6.20}$$

或使用(6.18)式有

$$b_{k+1} = b_k - c_k + 4^k. \tag{6.21}$$

最后有

$$c_{k+1} = a_k + 2c_k + d_k, \tag{6.22}$$

或

$$c_{k+1} = c_k - b_k + 4^k. \tag{6.23}$$

等式(6.19)、(6.21)和等式(6.23)可以共同用于计算任意希望的值 a_k．我们从下面的初始条件开始：$a_1=2$，$b_1=1$，$c_1=1$．从(6.19)式，(6.21)式和(6.23)式，我们可以计算：

$$a_2 = 2 \cdot 2 + 1 + 1 = 6,$$
$$b_2 = 1 - 1 + 4^1 = 4,$$

$$c_2 = 1 - 1 + 4^1 = 4.$$

通过列出相应的序列，很容易检验这些结果. 例如，{0，1，2，3}上长度为2且有偶数个0和偶数个3的六个序列是00，33，11，12，21，22. 类似地，可以得到

$$a_3 = 2 \cdot 6 + 4 + 4 = 20,$$
$$b_3 = 4 - 4 + 4^2 = 16,$$
$$c_3 = 4 - 4 + 4^2 = 16.$$

注意，我们没有找到单一的递推关系. 然而，我们找到了三个递推关系，联立地使用它们可以计算出所需的数.

本节练习

1. 假设对于$n \geq 1$，$a_n = 4a_{n-1} + 3$，且$a_1 = 5$. 求a_5和a_6.
2. 在例6.3中，令$r = 0.2$且$S_0 = \$5000$，并使用单利，使用其递推关系依次计算$S_1$，$S_2$，$S_3$，$S_4$，$S_5$和$S_6$，并通过使用等式$S_k = S_0(1 + kr)$来检验你的答案.
3. 当使用复利时，重复练习2，并使用等式$S_k = (1+r)^k S_0$来检验你的答案.
4. 在例6.4中，
   (a)推导a_4.
   (b)推导a_5.
   (c)通过列出长度为2的所有合法代码字，检证$a_2 = 10$.
   (d)对于$a_3 = 36$，重复(c).
5. 通过显示地列出{1，2，3}上的所有排列及把恒等排列转换成每一个排列所需的排列数量，对于b_3检验例6.6中的公式.
6. 对于例6.7，通过在平面上画出四条直线并标出区域来检验$f(4) = 11$. 355
7. 在初始条件$f(1) = 15$下，确定递推关系(6.5)式的解.
8. 确定递推关系(6.6)式的两个不同解.
9. 对于例6.8，计算b_4和b_5，并通过枚举出投4次和5次球且没有连续投中的所有方法检验你的答案.
10. 对于递推关系(6.13)式，确定一个不同于(6.14)式定义的序列的解.
11. 确定{1，2，3，4}的所有错位排列.
12. 对于6.1.4节的例子，使用递推关系(6.19)式，(6.21)式和(6.23)式计算a_4，b_4和c_4.
13. 第一天，把n项工作分配给n个工人. 第二天，再一次分配这n项工作，但是没有工人得到与他在第一天分配到的同样工作. 有多少种方法分配这两天的工作?
14. 在一次计算机系统检测中，一位银行雇员错误地删除了属于七个账户的七个"身份号码"记录. 在重新创建这些记录之后，他把那些身份号码随机地分配给这些账户. 他有多少种方法，使得：
    (a)至少一个身份号码得到正确的指定.
    (b)所有七个身份号码都得到正确的指定.
15. 一位实验室管理员可以做两种不同的实验，一个是开支很高的实验(E)花去8000美元，一个是开支不高的实验(I)花去4000美元. 例如，如果她有一笔12 000美元的预算，那么她可以按下面序列进行实验：III，IE或EI. 设$F(n)$是她正好花去n美元的实验序列数量. 因此$F(12\,000) = 3$.
    (a)确定$F(n)$的一个递推关系.
    (b)假设有p种不同的实验，E_1，E_2，$\cdots E_p$，且E_i要花销d_i美元. 在这种情况下确定$F(n)$的递推关系.
16. 假设我们有面值10分、18分和28分的邮票，每一种邮票都无限量供应. 如果我们考虑贴邮票的顺序，设$f(n)$是得到总邮资为n分的方法数量. 例如，$f(10) = 1$，$f(20) = 1$(两枚10分的邮票)，$f(28) = 3$(一枚28分的邮票，或者一枚10分的邮票加上一枚18分的邮票，或者一枚18分的邮票加上一枚10分的邮票).

(a)如果 $n>29$，推导出 $f(n)$的一个递推关系.

(b)使用(a)的递推关系，确定得到总邮资为 66 分的邮票的方法数量.

(c)通过给出所有的方法检验你对(b)的答案.

17. 一家工厂有两个部门. 假设在一周内，在第一部门中有 9 名工人被分配给 9 项不同的工作，在第二部门中另外 9 名工人被分配给 9 项不同的工作. 在接下的一周，管理人希望重新分配工作，使得工人都不会得到他原来的工作(在这家工厂每两项工作都是不同的).

356

(a)如果每一位工人都在工厂的相同部门，那么有多少种方法可以做到上面的分配?

(b)如果每一位工人要被转到不同于他们原来所在的工厂部门，那么有多少种方法可以做到上面的分配?

18. 在预测一种产品的未来销量时，一种(可能是不正确的)假设是说下一年的销售总量是这一年和上一年的销售总量的平均数. 假设 a_n 是第 n 年的销售总量.

(a)确定 a_n 的递推关系.

(b)如果 $a_0=a_1=1$，求解这一递推关系.

19. 确定{1，2，3，4，5，6，7，8}的错位排列数量：其中前四个元素必须映射成：

(a)以某个顺序排列的 1，2，3，4. (b)以某个顺序排列的 5，6，7，8.

20. 假设 $f(p)$是使用冒泡排序算法对 p 个项排序所需的比较数量(3.6.4 节). 确定 $f(p)$的递推关系并求解.

21. 数字表{0，1，2}上的代码字是合法的，当且仅当没有两个 0 连续出现. 确定长度为 n 的合法代码字的数量 b_n 的递推关系.

22. 数字表{0，1，2}上的代码字是合法的，当且仅当这个代码字有偶数个 0 和奇数个 1. 确定可以用于计算长度为 n 的合法代码字的数量的联立递推关系.

23. (a)使整数 1，2，…，9 中的每一个偶数都在各自的自然位置，而任意奇数都不放在其自然位置的排列有多少种?

(b)整数 1，2，…，9 的排列中正好有四个整数在其自然位置上的排列有多少种?

24. 在单打网球赛中，$2n$ 个队员被配对成 n 场比赛，且 $f(n)$是实现这种配对的不同方法的数量. 确定 $f(n)$的递推关系.

25. 假设有下面各 DNA 序列的随机排列. 原来的序列与排列后的序列之间的错误距离为 n 的概率是多少?

(a)AGT (b)AACC (c)ACTCGC (d)ACTGGGG (e)CTTAAA (f)CTAGG

26. 假设

$$a_n=\begin{cases}a_{n-1} & n\text{ 为偶数}\\ 2a_{n-2}+a_{n-4}+\cdots+a_3+a_1+1 & n\text{ 为奇数}\end{cases}$$

如果 $a_1=1$ 且 $a_3=3$，使用归纳法证明 $a_{2n}=a_{2n-1}=F_{2n-1}$，即 a_n 等于斐波那契数 $F_{2\lfloor\frac{n+1}{2}\rfloor-1}$.

27. (a)假设把椅子排成一个圆圈. 设 L_n 计数没有连续椅子的 n 把椅子子集的数量. 证明

$$L_{n+1}=F_n+F_{n-2}.$$

357

(数 L_n 称为卢卡斯数(Lucas number)[⊖].)

(b)确定 L_n 的两个初始条件，即 L_1 和 L_2.

(c)证明 $L_n=L_{n-1}+L_{n-2}$.

28. (a)证明：把 n 写成 1 和 2 的和的方法数量等于斐波那契数 F_n.

(b)只使用五分和一角硬币，你有多少种不同的方法把一美元投入一台自动贩卖机中?

29. 在我们的记法中，F_{n-1}是第 n 个斐波那契数，因为我们开始于 $F_0=1$.

⊖ Édouard Lucas(1842—1891)是一位法国数学家，是他把斐波那契在《Liber Abaci》(算盘书)中提出的兔子繁殖问题的序列解冠以斐波那契的名字.

(a)证明每第三个斐波那契数可被 $F_2=2$ 整除.

(b)证明每第四个斐波那契数都可被 $F_3=3$ 整除.

(c)证明每第五个斐波那契数都可被 $F_4=5$ 整除.

(d)证明每第 n 个斐波那契数都可被 F_{n-1} 整除.

30. 考虑斐波那契数 F_n，$n=0$，1，….

(a)证明 $F_{n-1}F_n-F_{n-2}F_{n+1}=(-1)^n$.

(b)证明 $F_{n+1}F_{n-1}-F_n^2=(-1)^n$.

(c)证明 $F_n=2+\sum_{k=1}^{n-2}F_k$.

(d)证明 $F_{2n+1}=1+\sum_{k=1}^{n}F_{2k}$.

(e)证明 $F_{n+1}=\sum_{k=0}^{\lfloor n/2\rfloor}\binom{n-k}{k}$.

(f)证明 $F_{n+m}=F_mF_{n+1}+F_{m-1}F_n$.

(g)证明 $F_{2n+1}=F_n^2+F_{n+1}^2$.

31. 如果 F_n 是第 n 个斐波那契数，确定下式的仅对于唯一一个 p 包含 Fp 的简单、闭式的表达式：

$$F_1+F_2+\cdots+F_n$$

32. 设 $S(n,t)$ 表示把 n 集合划分成 t 个非空无序子集的方法数量.⊖那么，对于 $t=1$，2，…，$n-1$，$S(n,t)$满足下式：

$$S(n,t)=tS(n-1,t)+S(n-1,t-1), \tag{6.24}$$

等式(6.24)是包含两个下标的递推关系的例子. 我们可以使用这一关系式计算任意的 $S(n,t)$. 例如，假设我们开始于下面的观察：对于所有的 n 有 $S(n,1)=1$，以及对于所的 n 有 $S(n,n)=1$. 于是，我们可以对于所有余下的 n 和 $t\leqslant n$，计算 $S(n,t)$. 我们可以根据 $S(2,2)$和 $S(2,1)$来计算 $S(3,2)$；可以根据 $S(3,2)$和 $S(3,1)$来计算 $S(4,2)$；根据 $S(3,3)$和 $S(3,2)$来计算 $S(4,3)$等等. [358]

(a)使用(6.24)式计算 $S(5,3)$.　　　　(b)计算 $S(6,3)$.

(c)证明(6.24)式成立.

33. 对于第2章的附加练习24，考虑有 m 条南北街和 n 条东西街的网格. 如果 A 和 B 是如图2.7所确定的位置，那么设 $s(m,n)$是从A到B的不同路线的数量. 确定 $s(m+1,n+1)$的一个递推关系.

34. 确定 $f(n)$的一个递推关系，$f(n)$是一个平面被 n 个圆划分的区域数量，其中每一对圆正好相交于两点且没有三个圆相交于一点.

35. (a)假设对于所有 $n\geqslant1$，$f(n+1)=f(n)f(n-1)$，且 $f(0)=f(1)=2$. 确定 $f(n)$.

(b)如果 $f(0)=f(1)=1$，重复(a).

36. 在例6.6中，我们计算了把恒等排列转换成{1，2，…，n}上随机选取的排列所需的转换数量 b_n. 这一计算是在这样的假设下进行的：{1，2，…，n}上的每一个排列被选择的几率是均等的. 对于恒等排列被选择几率是其他排列的两倍，且所有其他排列被选择的几率均等的情况，重新计算 b_n.

37. (Liu[1968])位串中的模式(pattern)是由若干连续数字组成的，例如，011. 我们说一个模式出现(occur)于一个位串的第 k 个数字，如果当从左到右扫描这一位串时，当第 k 个数字被扫描后这一完整的模式出现. 一旦一个模式出现，即发现一个模式，那么扫描(在其后)再一次开始. 例如，在位串110101010101中，模式010在第5个数字和第9个数字出现，而不在第7个数字出现. 设 b_n 表示长度为 n 且模式010出现于第 n 个数字的位串数量. 确定 b_n 的递推关系[提示：考虑结束于010的长度为 n 的位串数量，再把这些位串分成在第 n 个数字出现010模式和不在第 n 个数字出现010模式两类].

38. 假设 B 是 $n\times n$ 的棋盘，$r_n(B)$是车多项式 $R(x,B)$中 x^n 的系数. 使用递推关系计算 $r_n(B)$，如果：

⊖ $S(n,t)$被称为第二类斯特林数，我们已在2.10.4节和5.5.3节中讨论过.

(a)B的所有格子都是深色的.

(b)B只有主对角线是浅色的.

39. 在6.1.4节的例子中，只使用前面的a_p值确定a_k的单一递推关系.

40. 假设C_k是k个顶点连通标签图的数量. Harary和Palmer[1973]推导了如下递推关系：

$$C_k = 2^{\binom{k}{2}} - \frac{1}{k}\sum_{p=1}^{k-1}\binom{k}{p}C_p p 2^{\binom{k-p}{2}}.$$

使用事实$C_1=1$，计算C_2，C_3和C_4，并通过画出这些图检验你的答案.

41. 一个有p个0，q个1和r个2的序列是"好的"，如果在这一序列中没有连续的相同数字. 设$N(p, q, r)$是这样的"好"序列的数量.

359

(a)计算$N(p, q, 0)$.

(b)如果没有限制，那么有多少个p个0，q个1，r个2的不同序列.

(c)确定$N(p, q, r)$的递推关系.

6.2 特征根方法

6.2.1 不同根的情况

迄今为止，我们已经推导出若干有趣的递推关系. 对于其中的几个递推关系，我们可以通过迭代返回到原始值或初始条件来求解它们. 事实上，我们甚至可以对我们遇到过的更难的递推关系做类似的工作. 但是却没有求解所有递推关系的一般方法. 然而，存在适用于一大类递推关系的求解方法. 本节研究一个这样的方法，即特征根方法. 在6.3节，我们给出如何使用第5章所阐述的生成函数的概念求解递推关系. 求解递推关系的方法源自于差分方程理论. 差分方程的详细论述可以在Elaydi[1999]、Goldberg[1958]、Kelley和Peterson[2001]中找到.

考虑下面的递推关系：

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_p a_{n-p}, \tag{6.25}$$

其中，$n \geq p$，c_1，c_2，…，c_p是常数，且$c_p \neq 0$. 这样的一个递推关系称为是*线性的*(linear)，因为所有项a_k的次数都是一次幂，而且这样的递推关系称为是*齐次的*(homogeneous)[⊖]，因为右边任意一项都含有某个a_k，$n-p \leq k \leq n-1$. 因为系数c_i是常数，所以递推关系(6.25)式称为*常系数线性齐次递推关系*(linear homogeneous recurrence relation with constant coefficient). 递推关系(6.1)式、(6.7)式、(6.9)式和(6.11)式都是这样的递推关系的例子. 递推关系(6.4)式、(6.5)式、(6.6)式、(6.8)式和(6.10)式不是齐次的，递推关系(6.13)式不是常系数递推关系. 迄今为止我们所遇到的所有递推关系都是线性的.

我们将给出求解常系数线性齐次递推关系的一个技术. 这个技术非常类似于用于求解常系数线性微分方程的方法，这正如熟悉后者的读者将会看到的那样.

一旦我们指定了前p个项a_0，a_1，…，a_{p-1}的值，递推关系(6.25)式有唯一解. 这些值形成*初始条件*(initial condition). 根据a_0，a_1，…，a_{p-1}，我们能够使用这一递推关系确定a_p. 于是，根据a_1，…，a_{p-1}，a_p，我们能够确定a_{p+1}，以此类推.

一般地，如果忽略初始条件，那么递推关系(6.25)式有很多解. 其中的某些解将是形如下面的序列：

360

$$\alpha^0, \alpha^1, \alpha^2, \cdots, \alpha^n, \cdots, \tag{6.26}$$

其中α是一个数. 我们从确定使(6.26)式成为递推关系(6.25)式的一个解的α值开始. 在(6.25)式中，让我们用x^k替换a_k，并求解x. 做这一替换后，我们得到等式：

⊖ 前面我们已经在多项式的相关内容中定义了齐次的概念.

$$x^n - c_1x^{n-1} - c_2x^{n-2} - \cdots - c_px^{n-p} = 0. \tag{6.27}$$

(6.27)式的两边除以 x^{n-p}，我们得到

$$x^p - c_1x^{p-1} - c_2x^{p-2} - \cdots - c_p = 0. \tag{6.28}$$

等式(6.28)称为递推关系(6.25)式的**特征方程**(characteristic equation). 这个特征方程是一个 x 的 p 次多项式，所以有 p 个根 α_1，α_2，…，α_p. 其中的某些根可能是重根，某些可能是复数. 这些根被称为递推关系(6.25)式的**特征根**(characteristic root). 例如，考虑下面的递推关系：

$$a_n = 5a_{n-1} - 6a_{n-2}, \tag{6.29}$$

且有初始条件 $a_0=1$，$a_1=1$. 于是 $p=2$，$c_1=5$，$c_2=-6$，且特征方程由下式给出：

$$x^2 - 5x + 6 = 0.$$

这一特征方程有根 $x=2$ 和 $x=3$，所以 $\alpha_1=2$ 和 $\alpha_2=3$ 是特征根.

如果 α 是递推关系(6.25)式的特征根，而且如果我们取 $a_n=\alpha^n$，那么序列(a_n)满足这一递推关系. 因此，对应于每一个特征根，我们有这一递推关系的一个解. 在(6.29)式中，$a_n=2^n$ 和 $a_n=3^n$ 给出解. 然而，二者都不满足初始条件 $a_0=1$，$a_1=1$.

下面的重要发现是，如果序列(a_n')和(a_n'')都满足递推关系(6.25)式，而且如果 λ_1 和 λ_2 是常数，那么序列(a_n''')也是(6.25)式的解，其中 $a_n'''=\lambda_1a_n'+\lambda_2a_n''$. 换句话说，解的加权和还是解. 为了弄明白这一点，注意

$$a'_n = c_1a'_{n-1} + c_2a'_{n-2} + \cdots + c_pa'_{n-p} \tag{6.30}$$

及

$$a''_n = c_1a''_{n-1} + c_2a''_{n-2} + \cdots + c_pa''_{n-p}. \tag{6.31}$$

用 λ_1 乘以(6.30)式，λ_2 乘以(6.31)式，并把它们相加，得

$$\begin{aligned}
a'''_n &= \lambda_1a'_n + \lambda_2a''_n \\
&= \lambda_1(c_1a'_{n-1} + c_2a'_{n-2} + \cdots + c_pa'_{n-p}) + \lambda_2(c_1a''_{n-1} + c_2a''_{n-2} + \cdots + c_pa''_{n-p}) \\
&= c_1(\lambda_1a'_{n-1} + \lambda_2a''_{n-1}) + c_2(\lambda_1a'_{n-2} + \lambda_2a''_{n-2}) + \cdots + c_p(\lambda_1a'_{n-p} + \lambda_2a''_{n-p}) \\
&= c_1a'''_{n-1} + c_2a'''_{n-2} + \cdots + c_pa'''_{n-p}.
\end{aligned}$$

因此，a_n'''的确满足(6.25)式. 例如，如果我们定义 $a_n=3\cdot 2^n+8\cdot 3^n$，那么 a_n 满足(6.29)式. 361

一般地，假设 α_1，α_2，…，α_p 是递推关系(6.25)式的特征根. 那么我们的推理证明如果 λ_1，λ_2，…，λ_p 是常量，且如果有

$$a_n = \lambda_1\alpha_1^n + \lambda_2\alpha_2^n + \cdots + \lambda_p\alpha_p^n,$$

那么 a_n 满足(6.25)式. 可以证明，(6.25)式的每一个解都可以用这种形式表示，条件是特征根 α_1，α_2，…，α_p 是互不相同的. 对于这事实的证明，参见本小节的末尾.

定理 6.1 *假设常系数线性齐次递推关系(6.25)式有特征根 α_1，α_2，…，α_p. 那么，如果 λ_1，λ_2，…，λ_p 是常量，形如下面的每一个表达式是这一递推关系的一个解：*

$$a_n = \lambda_1\alpha_1^n + \lambda_2\alpha_2^n + \cdots + \lambda_p\alpha_p^n, \tag{6.32}$$

另外，如果这些特征根是互不相同的，那么这一递推关系的每一个解对于某些常量 λ_1，λ_2，…，λ_p 都有(6.32)式的形式.

我们称(6.32)式的表达式是递推关系(6.25)式的**一般解**(general solution).

根据定理 6.1 可以推定，为了寻找满足初始条件 a_0，a_1，…，a_{p-1} 的递推关系(6.25)式的唯一解，如果特征根是互不相同的，那么我们只需要确定一般解中的常量 λ_1，λ_2，…，λ_p 的值，使其满足这一初始条件. 我们来看如何确定这些常量 λ_i. 在(6.29)式中，每一个解都有形式

$$a_n = \lambda_12^n + \lambda_23^n.$$

现有我们有

$$a_0 = \lambda_1 2^0 + \lambda_2 3^0 \qquad a_1 = \lambda_1 2^1 + \lambda_2 3^1.$$

所以，根据 $a_0=1$，$a_1=1$，我们得到下面的方程组：

$$\lambda_1 + \lambda_2 = 1,$$
$$2\lambda_1 + 3\lambda_2 = 1.$$

这一方程组有唯一解 $\lambda_1=2$，$\lambda_2=-1$. 因此，因为 $\alpha_1 \neq \alpha_2$，所以满足给定初始条件的(6.29)式的唯一解是 $a_n = 2 \cdot 2^n - 3^n$.

一般过程的处理与本例相同. 如果通过(6.32)式来定义 a_n，我们使用 a_0，a_1，…，a_{p-1} 的初始值来建立关于 p 个未知数 λ_1，λ_2，…，λ_p 的 p 个联立方程组. 可以证明如果 α_1，α_2，…，α_p 互不相同，那么这一联立方程组总有一个唯一解. 这一事实的证明实际上是定理 6.1 余下部分的证明，我们现在给出其证明.

我们通过给出定理 6.1 陈述的一个大致证明来结束本小节，即如果特征根 α_1，α_2，…，α_p 是互不相同的，那么递推关系(6.25)式的每一个解对于某些常量 λ_1，λ_2，…，λ_p[⊖]都有(6.32)式的形式. 假设

362

$$b_n = \lambda_1 \alpha_1^n + \lambda_2 \alpha_2^n + \cdots + \lambda_p \alpha_p^n$$

是(6.25)式的一个解. 使用初始条件 $b_0=a_0$，$b_1=a_1$，…，$b_{p-1}=a_{p-1}$，我们有

$$\left.\begin{aligned} \lambda_1 + \lambda_2 + \cdots + \lambda_p &= a_0 \\ \lambda_1\alpha_1 + \lambda_2\alpha_2 + \cdots + \lambda_p\alpha_p &= a_1 \\ \lambda_1\alpha_1^2 + \lambda_2\alpha_2^2 + \cdots + \lambda_p\alpha_p^2 &= a_2 \\ &\vdots \\ \lambda_1\alpha_1^{p-1} + \lambda_2\alpha_2^{p-1} + \cdots + \lambda_p\alpha_p^{p-1} &= a_{p-1}. \end{aligned}\right\} \tag{6.33}$$

方程组(6.33)是关于 p 个未知数 λ_1，λ_2，…，λ_p 的 p 个线性方程组. 现在考虑该方程组(6.33)式的系数矩阵.

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_p \\ \alpha_1^2 & \alpha_2^2 & \cdots & \alpha_p^2 \\ \vdots & \vdots & \cdots & \vdots \\ \alpha_1^{p-1} & \alpha_2^{p-1} & \cdots & \alpha_p^{p-1} \end{bmatrix}.$$

这一矩阵的行列式就是著名的范德蒙德行列式(Vandermonde determinant). 可以证明，范德蒙德行列式是由下面所有项 $\alpha_j - \alpha_i$ 的积给出的，其中 $1 \leqslant i < j \leqslant p$：

$$\prod_{1 \leqslant i < j \leqslant p} (\alpha_j - \alpha_i),$$

因为 α_1，α_2，…，α_p 是互不相同的，所以行列式不等于 0. 因此，(6.33)式的方程组存在唯一解 λ_1，λ_2，…，λ_p. 因此，我们看到有初始条件 a_0，a_1，…，a_{p-1} 的递推关系(6.25)式有形如(6.32)式的解. 现在，有这些初始条件的递推关系只有一个解，因此它必须是这样的形式. 这就是完整的证明.

6.2.2 第 k 个斐波那契数的计算

让我们使用另外一个例子说明这一方法，即斐波那契数的递推关系(6.11)式，在这里我们重

⊖ 本证明可以跳过.

复这一递推关系：

$$F_k = F_{k-1} + F_{k-2}. \tag{6.34}$$

这里，$p=2$ 且 $c_1=c_2=1$. 特征方程是 $x^2-x-1=0$. 根据二次公式，这一方程的根，即特征根是 $\alpha_1=(1+\sqrt{5})/2$，$\alpha_2=(1-\sqrt{5})/2$. 因为 $\alpha_1 \neq \alpha_2$，所以一般解是

$$\lambda_1\left(\frac{1+\sqrt{5}}{2}\right)^k + \lambda_2\left(\frac{1-\sqrt{5}}{2}\right)^k.$$

初始条件 $F_0=F_1=1$ 给出我们两个方程：

363

$$\lambda_1 + \lambda_2 = 1,$$

$$\lambda_1\left(\frac{1+\sqrt{5}}{2}\right) + \lambda_2\left(\frac{1-\sqrt{5}}{2}\right) = 1.$$

求解关于 λ_1 和 λ_2 的联立方程组得到

$$\lambda_1 = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right), \quad \lambda_2 = -\frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right),$$

因此，在给定的初始条件下，(6.34)式的解，即第 k 个斐波那契数由下式给出：

$$F_k = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)\left(\frac{1+\sqrt{5}}{2}\right)^k - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)\left(\frac{1-\sqrt{5}}{2}\right)^k,$$

或

$$F_k = \frac{\left(\frac{1+\sqrt{5}}{2}\right)^{k+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{k+1}}{\sqrt{5}}, \tag{6.35}$$

或

$$F_k = \frac{\tau^{k+1} - (1-\tau)^{k+1}}{\sqrt{5}},$$

其中 τ 就是 6.1.2 节的黄金比例. 在 6.3 节的练习 8 中，我们使用生成函数推导这一结果.

【例 6.12 信息传输期间(再探例 6.5)】 我们注意到递推关系(6.9)式和(6.34)式是相同的. 另外，它们的初始条件也是相同的. 因为 $F_1=1$，$F_2=2$，而 $N_1=1$ 和 $N_2=2$. 又因为相同原因，我们取 F_0 等于 0，即为了维持这一递推关系对于 $k=2$ 也成立，我们取 N_0 等于 0. 现在，正如我们早前所观察的那样，如果我们已知有互不相同的特征根和初始条件 a_0，a_1，…，a_{p-1} 的递推关系(6.25)式，那么其解唯一确定. 因此，由此得出对于所有 $t \geqslant 0$，N_t 必须等于 F_t，所以我们可以使用(6.35)式计算 N_k. 根据这一结果不难证明例 6.5 定义的 Shannon 容量可以由下式给出：

$$C = \log_2\left(\frac{1+\sqrt{5}}{2}\right).$$

■

6.2.3 重根的情况

考虑递推关系

$$a_n = 6a_{n-1} - 9a_{n-2}, \tag{6.36}$$

其中 $a_0=1$，$a_1=2$. 其特征方程是 $x^2-6x+9=0$，或 $(x-3)^2=0$. 两个特征根是 3 和 3，即 3 是重根. 因此，不能运用定理 6.1 的第二部分. 尽管 3^n 是(6.36)式的解还成立，且 $\lambda_1 3^n + \lambda_2 3^n$ 总是一个解也成立，但是(6.36)式的每一个解都有 $\lambda_1 3^n + \lambda_2 3^n$ 的形式不再成立. 特别地，不存在这样的满足我们所给初始条件的解. 因为这些条件给出我们下面的方程：

364

$$\lambda_1 + \lambda_2 = 1,$$

$$3\lambda_1 + 3\lambda_2 = 2.$$

不存在满足这两个方程的 λ_1 和 λ_2.

假设 α 是重数为 u 的特征根，即作为特征方程的根它出现正好 u 次. 那么，事实表明，不仅 $a_n=\alpha^n$ 满足递推关系(6.25)式，而且 $a_n=n\alpha^n$，$a_n=n^2\alpha^n\cdots$，$a_n=n^{u-1}\alpha^n$ 也满足(6.25)式(参见练习 29 和练习 30). 在我们上面的例子中，3 是重数 $u=2$ 的特征根，所以 $a_n=3^n$ 和 $a_n=n3^n$ 都是(6.25)式的解. 另外，因为解的加权和仍然是解，且因为 $a_n'=3^n$，$a_n''=n3^n$ 都是解，所以 $a_n'''=\lambda_1 3^n+\lambda_2 n3^n$ 也是解. 使用 a'''_n 的表达式和初始条件 $a_0=1$ 和 $a_1=2$，我们得到方程

$$\lambda_1 = 1,$$
$$3\lambda_1 + 3\lambda_2 = 2.$$

这一联立方程组有唯一解 $\lambda_1=1$，$\lambda_2=-\frac{1}{3}$. 因此，$a_n=3^n-\frac{1}{3}\cdot n\cdot 3^n$ 是有初始条件 $a_0=1$ 且 $a_1=2$ 的递推关系(6.36)式的一个解. 因此这是唯一解.

这一过程扩展如下. 假设递推关系(6.25)式有特征根 α_1，α_2，…，α_q，且特征根 α_i 的重数为 u_i. 那么下列各项：

$$\alpha_1^n, n\alpha_1^n, n^2\alpha_1^n, \cdots, n^{u_1-1}\alpha_1^n, \alpha_2^n, n\alpha_2^n, n^2\alpha_2^n, \cdots, n^{u_2-1}\alpha_2^n, \cdots,$$
$$\alpha_q^n, n\alpha_q^n, n^2\alpha_q^n, \cdots, n^{u_q-1}\alpha_q^n$$

必定都是这一递推关系的解. 让我们称这些解为**基础解**(basic solution). 总共有 p 个基础解. 我们把 p 个基础解分别记为 b_1，b_2，…，b_p. 因为解的加权和仍然为解，对于任意常量 λ_1，λ_2，…，λ_p，下面的表达式：

$$a_n = \lambda_1 b_1 + \lambda_2 b_2 + \cdots + \lambda_p b_p$$

也是这一递推关系的解. 根据用于定理 6.1 的证明的类似方法，可以证明，对于某些常量 λ_1，λ_2，…，λ_p，这一递推关系的每一个解都有这样的形式.

定理 6.2 *假设常系数线性齐次递推关系(6.25)式有基础解 b_1，b_2，…，b_p. 那么，对于某些常量 λ_1，λ_2，…，λ_p，其一般解由下式给出：*

365

$$a_n = \lambda_1 b_1 + \lambda_2 b_2 + \cdots + \lambda_p b_p. \tag{6.37}$$

满足初始条件 a_0，a_1，…，a_{p-1} 的唯一解可以通过在(6.37)式中设 $n=0$，1，…，$p-1$，并得到关于 p 个未知数 λ_1，λ_2，…，λ_p 的 p 个联立方程而进行计算.

为了说明这一计算过程，考虑递推关系

$$a_n = 7a_{n-1} - 16a_{n-2} + 12a_{n-3}, \tag{6.38}$$

$a_0=1$，$a_1=2$，$a_2=0$. 那么特征方程是

$$x^3 - 7x^2 + 16x - 12 = 0,$$

因式分解成 $(x-2)(x-2)(x-3)=0$. 所以特征根是 $\alpha_1=2$，其重数 $u_1=2$，$\alpha_2=3$，其重数 $u_2=1$. 因此，(6.38)式的一般解有下面的形式：

$$a_n = \lambda_1\alpha_1^n + \lambda_2 n\alpha_1^n + \lambda_3\alpha_2^n = \lambda_1 2^n + \lambda_2 n2^n + \lambda_3 3^n.$$

令 $n=0$，1，2，我们得

$$a_0 = \lambda_1 + \lambda_3 = 1,$$
$$a_1 = 2\lambda_1 + 2\lambda_2 + 3\lambda_3 = 2,$$
$$a_2 = 4\lambda_1 + 8\lambda_2 + 9\lambda_3 = 0.$$

这一方程组的唯一解是 $\lambda_1=5$，$\lambda_2=2$，$\lambda_3=-4$. 因此，有给定初始条件的(6.38)式的唯一解是

$$a_n = 5\cdot 2^n + 2\cdot n2^n - 4\cdot 3^n.$$

本节练习

1. 下列递推关系中哪些是线性的？

(a)$a_n=5a_{n-1}+2a_{n-2}+3$
(b)$b_n=3b_{n-1}+9b_{n-2}+18b_{n-3}+32b_{n-4}$
(c)$c_n=21c_{n-2}+4c_{n-5}$
(d)$d_n=16d_{n-1}-12d_{n-2}$
(e)$e_n=24e_{n-1}+22e_{n-2}^2$
(f)$f_n=nf_{n-1}+f_{n-2}$
(g)$g_n=n^2g_{n-2}$
(h)$h_n=8h_{n-3}+81$
(i)$i_n=5i_{n-1}+3^n i_{n-2}$

2. 练习1的递推关系中哪些是齐次的？
3. 练习1的递推关系中哪些是常系数的？
4. 使用(6.35)式对于$k=2, 3, 4, 5, 6, 7$推导F_k的值.
5. 确定下列每一个递推关系的特征方程.

(a)$a_n=-2a_{n-1}-a_{n-2}$
(b)$b_k=-7b_{k-1}+18b_{k-2}$
(c)$c_n=3c_{n-1}+18c_{n-2}-7c_{n-3}$
(d)$d_n=81d_{n-4}+4d_{n-5}$
(e)$e_k=4e_{k-2}$
(f)$f_{n+1}=2f_n+3f_{n-1}$
(g)$g_n=18g_{n-7}$
(h)$h_n=9h_{n-2}$
(i)$i_n=i_{n-1}+4i_{n-2}-4i_{n-3}$
(j)$j_n=2j_{n-1}+9j_{n-2}-18j_{n-3}$
(k)$k_n=11k_{n-1}+22k_{n-2}+11k_{n-3}-33k_{n-8}$

366

6. 对于练习5，确定(a)、(b)、(e)、(f)、(h)、(i)和(j)的递推关系的特征根. [提示：2是(i)和(j)的根.]
7. (a)证明递推关系$a_n=5a_{n-1}$可以有多个解.

(b)证明如果我们知道$a_0=20$，那么这一递推关系有唯一解.

8. 使用特征根方法求解下列递推关系.

(a)$a_n=6a_{n-1}$，$a_0=5$
(b)$t_{k+1}=2t_k$，$t_1=1$(这是例6.1)

9. 考虑递推关系$a_n=15a_{n-1}-44a_{n-2}$. 证明下列各序列是解.

(a)(4^n)
(b)$(3\cdot 11^n)$
(c)(4^n-11^n)
(d)(4^{n+1})

10. 对于练习9，下列各序列哪些是解？

(a)(-4^n)
(b)(4^n+1)
(c)$(3\cdot 4^n+12\cdot 11^n)$
(d)$(n4^n)$
(e)$(4^n 11^n)$
(f)$((-4)^n)$

11. 考虑递推关系$b_k=7b_{k-1}-10b_{k-2}$. 下列各序列哪些是解？

(a)(2^k)
(b)(5^k-2^k)
(c)(2^k+7)
(d)(2^k-5^k)
(e)(2^k+5^{k+1})

12. 使用特征根方法并在下面的初始条件下，求解练习5中的下列递推关系.

(a)练习5(a)，$a_0=2$，$a_1=2$
(b)练习5(b)，$b_0=0$，$b_1=8$
(c)练习5(e)，$e_0=-1$，$e_1=1$
(d)练习5(f)，$f_0=f_1=2$
(e)练习5(h)，$h_0=4$，$h_1=2$
(f)练习5(i)，$i_0=0$，$i_1=1$，$i_2=2$
(g)练习5(j)，$j_0=2$，$j_1=1$，$j_2=0$

13. 假设在例6.5中，a的传输需要2个单位时间，而b的传输需要3个单位时间. 求解N_t.
14. 求解6.1节练习18中的产品销售问题中的a_n，假设$a_0=0$，$a_1=1$.
15. 使用(6.35)式，检证6.1节练习29的结果.
16. 使用(6.35)式，检证6.1节练习30的结果.
17. 考虑递推关系$a_n=-a_{n-2}$.

(a)证明i和$-$i是特征根(注意：$\mathrm{i}=\sqrt{-1}$).

(b)序列(i^n)是解吗？

(c)序列$(2\mathrm{i}^n+(-\mathrm{i})^n)$又如何？

(d)假设 $a_0=0$，$a_1=1$，确定这一递推关系的唯一解.

367

18. 考虑递推关系 $a_n=-9a_{n-2}$

(a)确定一般解.

(b)假设 $a_0=0$，$a_1=5$，确定这一递推关系的唯一解.

19. 考虑递推关系 $F_n=4F_{n-1}-4F_{n-2}$. 证明：存在一个多重特征根 α，并证明对于初始条件 $F_0=1$，$F_1=3$，不存在常量 λ_1 和 λ_2，使得对于所有的 n 都有 $F_n=\lambda_1\alpha^n+\lambda_2\alpha^n$.

20. 假设(a_n')和(a_n'')是递推关系(6.25)式的两个解，且 $a_n'''=a_n'-a_n''$. 那么，a_n'''一定是(6.25)式的解吗？为什么？

21. 假设(a_n')，(a_n'')和(a_n''')是递推关系(6.25)式的三个解，且我们有 $b_n=\lambda_1 a_n'+\lambda_2 a_n''+\lambda_3 a_n'''$. (b_n)一定是(6.25)式的解吗？为什么？

22. 考虑递推关系

$$a_n=9a_{n-1}-27a_{n-2}+27a_{n-3}.$$

证明下列每一个序列都是它的解.

(a)(3^n)　　(b)$(n3^n)$　　(c)$(n^2 3^n)$

(d)$(3\cdot 3^n)$　　(e)(3^n+n3^n)　　(f)$(4\cdot 3^n+8\cdot n3^n-n^2 3^n)$

23. 考虑递推关系

$$a_n=3a_{n-1}-3a_{n-2}+a_{n-3}.$$

证明下列各序列都是它的解.

(a)(1，1，1，…)　　(b)(0，1，2，3，…)　　(c)(0，1，4，9，…)

24. 考虑递推关系

$$b_n=9b_{n-1}-24b_{n-2}+20b_{n-3}.$$

证明序列(2^n)，$(n2^n)$和(5^n)都是它的解.

25. 考虑递推关系

$$c_k=13c_{k-1}-60c_{k-2}+112c_{k-3}-64c_{k-4}.$$

证明下列各序列都是它的解.

(a)(2，2，…)　　(b)$(3\cdot 4^k)$　　(c)$(4^k+k4^k+k^2 4^k)$

(d)(4^k+1)　　(e)$(k4^k-11)$

26. 确定下列各递推关系在给定初始条件下的唯一解.

(a)练习 22 的递推关系，初始条件为 $a_0=0$，$a_1=1$，$a_2=1$.

(b)练习 23 的递推关系，初始条件为 $a_0=1$，$a_1=1$，$a_2=2$.

(c)练习 24 的递推关系，初始条件为 $b_0=1$，$b_1=2$，$b_2=0$.

(d)练习 25 的递推关系，初始条件为 $c_0=c_1=0$，$c_2=10$，$c_3=0$.

27. 在给定初始条件下求解下列各递推关系.

368

(a)$a_n=10a_{n-1}-25a_{n-2}$，$a_0=1$，$a_1=2$

(b)$b_k=14b_{k-1}-49b_{k-2}$，$b_0=0$，$b_1=10$

(c)$c_n=9c_{n-1}-15c_{n-2}+7c_{n-3}$，$c_0=0$，$c_1=1$，$c_2=2$[提示：$x=1$ 是一个特征根].

(d)$d_n=13d_{n-1}-40d_{n-2}+36d_{n-3}$，$d_0=d_1=1$，$d_2=0$[提示：$x=2$ 是一个特征根].

(e)$e_k=10e_{k-1}-37e_{k-2}+60e_{k-3}-36e_{k-4}$，$e_0=e_1=e_2=0$，$e_3=5$[提示：$x=2$ 和 $x=3$ 是特征根].

28. 在给定初始条件下求解下面的递推关系.

$$a_n=-2a_{n-2}-a_{n-4},\quad a_0=0,\quad a_1=1,\quad a_2=2,\quad a_3=3.$$

[提示：$x=i$ 和 $x=-i$ 是特征根.]

29. 假设 α 是递推关系(6.25)式的特征根，且 α 有重数 2. 证明(α^n)和$(n\alpha^n)$是(6.25)式的解[提示：如果 $C(x)=0$ 是特征方程，那么对于某个多项式 $D(x)$，$C(x)=(x-\alpha)^2D(x)$. 如果 $C_n(x)=x^{n-p}C(x)$，证明 α 是导数 $C_n'(x)$的根. 使用 α 替换方程 $xC_n'(x)=0$ 中的 x，证明$(n^2\alpha^n)$是(6.25)式的解].

30. (a)假设 α 是递推关系(6.25)式的特征根，且 α 的重数为 3. 证明(α^n)、$(n\alpha^n)$和$(n^2\alpha^n)$是(6.25)式的解.

[提示：注意 $C(x)=(x-\alpha)^3D(x)$ 这一事实，扩展练习 29 的论证. 考虑 $C_n(x)=x^{n-p}C(x)$，$A_n(x)=xC_n'(x)$ 和 $B_n(x)=xA_n'(x)$. 通过考虑 $A_n(x)=0$，证明 $(n^2\alpha^n)$ 是解，通过考虑 $B_n(x)=0$ 证明 $(n\alpha^n)$ 是解.]

(b)对于 α 是重数 u 的特征根，扩展上面的结果.

6.3 使用生成函数求解递推关系

6.3.1 方法

求解递推关系的另一个方法是使用第 5 章所讨论的生成函数的概念. 假设 $G(x)$ 是序列 (a_k) 的普通生成函数，即

$$G(x)=\sum_{k=0}^{\infty}a_kx^k.$$

我们尝试通过确定它的生成函数来确定 (a_k). 特别地，如果我们有 a_k 的递推关系，这一技巧就是使用 x^k 乘以这一递推关系的两边，然后取和，这给出用于推导 $G(x)$ 的表达式.

【例 6.13 小麦数量(再探例 6.1)】 我们利用例 6.1 的递推关系说明这一方法：

$$t_{k+1}=2t_k. \tag{6.39}$$

369

初始条件是 $t_1=1$. 对于这一情况，t_0 没有定义. 然而，通常把我们的序列考虑成从第 0 项开始比较便利. 因此，如果这一序列的前项是未知或没有给出，我们就尝试着根据递推关系定义它们. 特别地，下面的定义符合(6.39)式：

$$t_0=\frac{1}{2}t_1=\frac{1}{2}.$$

(t_k) 的普通生成函数是

$$G(x)=\sum_{k=0}^{\infty}t_kx^k.$$

为了推导 $G(x)$，我们从 x^k 乘以递推关系(6.39)式的两边开始，得到

$$t_{k+1}x^k=2t_kx^k.$$

于是，我们求和[⊖]，得到

$$\sum_{k=0}^{\infty}t_{k+1}x^k=2\sum_{k=0}^{\infty}t_kx^k. \tag{6.40}$$

(6.40)式的右边是 $2G(x)$. 那么它的左边是什么呢？我们尝试把(6.40)式化简成包含 $G(x)$ 的表达式. 注意

$$\begin{aligned}\sum_{k=0}^{\infty}t_{k+1}x^k&=t_1+t_2x+t_3x^2+\cdots\\&=\frac{1}{x}[t_1x+t_2x^2+t_3x^3+\cdots]\\&=\frac{1}{x}[t_0+t_1x+t_2x^2+t_3x^3+\cdots]-\frac{1}{x}t_0\\&=\frac{1}{x}[G(x)-t_0].\end{aligned}$$

因此

$$\sum_{k=0}^{\infty}t_{k+1}x_k=\frac{G(x)-t_0}{x}. \tag{6.41}$$

⊖ 这个求和是对递推关系成立的所有 k 的值求和. 在某些情况下，从 $k=1$ 或 $k=2$ 等开始求和更好或更合适.

等式(6.40)和(6.41)给出等式：

$$\frac{G(x)-t_0}{x}=2G(x).$$

这一等式，即 $G(x)$ 的函数等式(functional equation)可以用于求解 $G(x)$. 稍微使用一点代数变换可以得到

370

$$G(x)=\frac{t_0}{1-2x}.$$

因为我们已计算出 $t_0=\frac{1}{2}$，所以我们有

$$G(x)=\frac{1}{2}(1-2x)^{-1}.$$

知道了 $G(x)$，我们可以根据它计算出想要的 t_k 值. 如果我们展开 $G(x)$，那么数 t_k 就是其中的 x^k 的系数. 我们又如何展开 $G(x)$ 呢？有两个方法. 最容易的一种方法是对于 $|y|<1$ 利用恒等式[参见等式(5.2)]：

$$1+y+y^2+\cdots+y^n+\cdots=\frac{1}{1-y}, \tag{6.42}$$

这样做给出下面的结果：

$$G(x)=\frac{1}{2}[1+(2x)+(2x)^2+\cdots+(2x)^n+\cdots],$$

或

$$G(x)=\frac{1}{2}+x+2x^2+\cdots+2^{n-1}x^n+\cdots.$$

换句话说，

$$t_k=2^{k-1},$$

这与我们早前的计算相同. 展开 $G(x)$ 的另一种方法是使用二项式定理(定理 5.3). 我们把这一方法的具体形式留给读者自己去完成. ■

【例 6.14 合法代码字(再探例 6.4)】 我们利用下面给出的例 6.4 的递推关系说明这一方法：

$$a_{k+1}=2a_k+4^k. \tag{6.43}$$

使用其普通生成函数

$$G(x)=\sum_{k=0}^{\infty}a_kx^k.$$

我们知道 $a_1=3$. 根据这一递推关系，可以推导出 a_0，即使 a_0 没有定义. 我们得到

$$\begin{aligned}a_1&=2a_0+4^0,\\3&=2a_0+1,\\a_0&=1.\end{aligned}$$

现在，我们使用 x^k 乘以递推关系(6.43)式的两边并求和，得到

371

$$\sum_{k=0}^{\infty}a_{k+1}x^k=2\sum_{k=0}^{\infty}a_kx^k+\sum_{k=0}^{\infty}4^kx^k. \tag{6.44}$$

(6.44)式的左边由下式给出：

$$\frac{1}{x}\sum_{k=0}^{\infty}a_{k+1}x^{k+1}=\frac{1}{x}[G(x)-a_0]=\frac{1}{x}[G(x)-1].$$

因此，我们得到

$$\frac{1}{x}[G(x)-1]=2G(x)+\sum_{k=0}^{\infty}(4x)^k. \qquad (6.45)$$

根据恒等式(6.42)，我们重写上式为

$$\frac{1}{x}[G(x)-1]=2G(x)+\frac{1}{1-4x}.$$

根据这一函数等式，通过简单的代数变换可求解 $G(x)$. 我们得到

$$G(x)-1=2xG(x)+\frac{x}{1-4x},$$

$$G(x)(1-2x)=1+\frac{x}{1-4x},$$

$$G(x)=\frac{1}{1-2x}\left(1+\frac{x}{1-4x}\right),$$

$$G(x)=\frac{1}{1-2x}+\frac{x}{(1-2x)(1-4x)}. \qquad (6.46)$$

上式给出(a_k)的生成函数. 我们如何确定 a_k 呢？展开(6.46)式右边的第一项就足够了. 我们可以通过部分分数方法展开第二项.[⊖]即对于适当的 a 和 b，右边的第二项可以表示成如下形式：

$$\frac{a}{1-2x}+\frac{b}{1-4x}.$$

我们计算得到

$$a=-\frac{1}{2},\quad b=\frac{1}{2}.$$

因此，

$$G(x)=\frac{1}{1-2x}+\frac{-\frac{1}{2}}{1-2x}+\frac{\frac{1}{2}}{1-4x}=\frac{\frac{1}{2}}{1-2x}+\frac{\frac{1}{2}}{1-4x}. \qquad (6.47)$$

为了展开(6.47)式，我们再一次使用恒等式(6.42)，得到

$$G(x)=\frac{1}{2}\sum_{k=0}^{\infty}(2x)^k+\frac{1}{2}\sum_{k=0}^{\infty}(4x)^k.$$

372

因此，x^k 的系数由下式给出：

$$a_k=\frac{1}{2}(2)^k+\frac{1}{2}(4)^k.$$

特别地，我们可以检验 6.1 节中的计算. 我们有

$$a_2=\frac{1}{2}(2)^2+\frac{1}{2}(4)^2=10,$$

$$a_3=\frac{1}{2}(2)^3+\frac{1}{2}(4)^3=36.$$

读者也许希望用另一种方法检验我们的结果，即通过 5.5 节的方法直接计算 a_k 的指数生成函数.

■

【例 6.15 车多项式[⊜]】 在例 5.10 和例 5.14 中，我们介绍了车多项式，并陈述了把棋盘 B 的车多项式的计算化简成“更简单的”棋盘车多项式的计算的一个结果. 这里我们陈述另一个这样的结果. 假设 s 是棋盘 B 的任意一个深色格. 设 B_s 是由 B 通过禁止 s(使其成为浅色)得到的，

⊖ 对此方法的讨论参见大多数微积分教材.

⊜ 如果读者跳过了第 5 章，那么本例可以跳过.

而且设B_s'是由B通过禁止所有格子与s同行或同列而得到的. 图6.7给出一个棋盘B，一个格子s，以及棋盘B_s和B_s'.

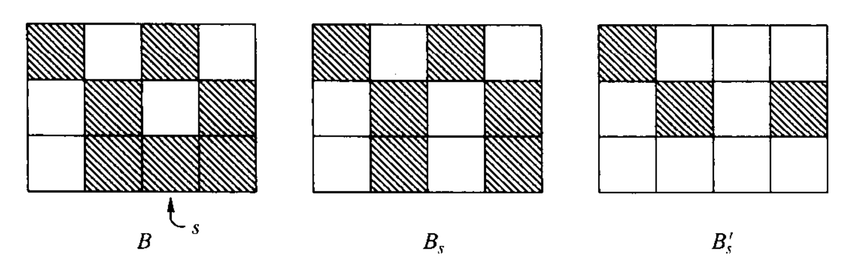


图6.7 一个棋盘B，一个格子s[(3，3)的格子]，以及由棋盘B和格子s得到的棋盘B_s和B_s'

注意，为了把$k\geqslant 1$个车放置在B上，我们或者使用格子s或者不使用格子s. 如果我们不使用格子s，我们必须把k个车放置在B_s的格子上. 如果我们使用格子s，那么我们仍有$k-1$个车需要放置，我们可以使用除那些与s在同一行或同一列之外的B的任意深色格子；即我们也可以使用B_s'中的任意深色格子. 因此，根据第2章的加法规则，对于$k\geqslant 1$，我们有下式：

$$r_k(B)=r_k(B_s)+r_{k-1}(B'_s). \tag{6.48}$$

如果我们用x^k乘以(6.48)式的两边，并对所有$k\geqslant 1$求和，我们求得

373

$$\sum_{k=1}^{\infty} r_k(B)x^k=\sum_{k=1}^{\infty} r_k(B_s)x^k+\sum_{k=1}^{\infty} r_{k-1}(B'_s)x^k. \tag{6.49}$$

(6.49)式的左边的项正好是

$$R(x,B)-r_0(B)=R(x,B)-1,$$

因为对于所有B，有$r_0(B)=1$. (6.49)式右边的第一项是

$$R(x,B_s)-r_0(B_s)=R(x,B_s)-1.$$

而其右边第二项等于

$$x\sum_{k=1}^{\infty} r_{k-1}(B'_s)x^{k-1}=x\sum_{k=0}^{\infty} r_k(B'_s)x^k=xR(x,B'_s).$$

因此，(6.49)式给出下式：

$$R(x,B)-1=R(x,B_s)-1+xR(x,B'_s),$$

或

$$R(x,B)=R(x,B_s)+xR(x,B'_s). \tag{6.50}$$

把这一结果应用于图6.7的棋盘B留作练习(练习27). ■

前面三个例子所使用的方法可以用于求解各种递推关系. 对于类似于(6.25)式这样的常系数线性齐次递推关系总是可行的⊖. 其结果将给出如下形式的生成函数$G(x)$：

$$\frac{p(x)}{q(x)},$$

其中$p(x)$是阶小于p的多项式，而$q(x)$是阶为p且常数项等于1的多项式[多项式$p(x)$和$q(x)$可以用递推关系(6.25)式的系数c_1，c_2，…，c_p和初始条件a_0，a_1，…，a_{p-1}来表示. 对于细节可以参见Brualdi[1999]或练习25]. 如果$q(x)$的所有根都是实数，那么可以使用部分分数方法把$p(x)/q(x)$表示成形如下面的项的和：

$$\frac{\alpha}{(1-\beta x)^t},$$

⊖ 初次阅读时本小节的余下内容可以跳过.

其中 t 是正整数，α 和 β 是实数. 于是，可以使用二项式定理展开项

$$\frac{\alpha}{(1-\beta x)^t}$$

得到

[374]

$$\frac{\alpha}{(1-\beta x)^t}=\alpha\sum_{k=0}^{\infty}\binom{t+k-1}{k}\beta^k x^k. \tag{6.51}$$

也可以使用 βx 取代推论 5.4.1 中的 x 来直接得到这一结果.

如果 $q(x)$ 有复根，那么可以使用部分分数法把 $p(x)/q(x)$ 表示成如下形式的项的和：

$$\frac{a}{(x-b)^t} \quad 或 \quad \frac{ax+b}{(x^2+cx+d)^t},$$

其中 t 是正整数，a，b，c 和 d 是实数. 前者的项可以变成下面的形式：

$$\frac{\alpha}{(1-\beta x)^t}.$$

而后者的项可以通过对分母做完全平方，然后再使用下式的展开式来加以处理：

$$\frac{1}{(1+y^2)^t}=\frac{1}{[1-(-y^2)]^t}.$$

我们省略其细节.

6.3.2 错位排列

下面，我们使用本节的技巧推导 n 元素错位排列的数量 D_n 的公式. 我们有下面的递推关系：

$$D_{n+1}=n(D_{n-1}+D_n), \tag{6.52}$$

其中，$n\geqslant 2$. 我们已知 $D_2=1$ 和 $D_1=0$. 因此，使用递推关系(6.52)式，我们推导出 $D_0=1$. 因为 $D_0=1$，(6.52)式对 $n\geqslant 1$ 成立. 递推关系(6.52)式不是很方便，因为它用 D_n 和 D_{n-1} 表示 D_{n+1}. 使用某种代数处理可以把(6.52)式化简成下面的递推关系：

$$D_{n+1}=(n+1)D_n+(-1)^{n+1}, \tag{6.53}$$

其中，$n\geqslant 0$. 对于这一事实的详细论证，请看本小节的最后部分.

让我们尝试着计算下面的普通生成函数：

$$G(x)=\sum_{n=0}^{\infty}D_n x^n.$$

我们用 x^n 乘以(6.53)式并求和，得到下式：

$$\sum_{n=0}^{\infty}D_{n+1}x^n=\sum_{n=0}^{\infty}(n+1)D_n x^n+\sum_{n=0}^{\infty}(-1)^{n+1}x^n. \tag{6.54}$$

(6.54)式的左边是

[375]

$$\frac{1}{x}[G(x)-D_0]=\frac{1}{x}G(x)-\frac{1}{x}.$$

使用恒等式(6.42)，(6.54)式右边的第二项变成

$$-\sum_{n=0}^{\infty}(-1)^n x^n=-\sum_{n=0}^{\infty}(-x)^n=-\frac{1}{1+x},$$

最后，(6.54)式右边的第一项可以重写成如下形式：

$$\sum_{n=0}^{\infty}nD_n x^n+\sum_{n=0}^{\infty}D_n x^n=x\sum_{n=0}^{\infty}nD_n x^{n-1}+\sum_{n=0}^{\infty}D_n x^n=xG'(x)+G(x).$$

因此，(6.54)式变成

$$\frac{1}{x}G(x)-\frac{1}{x}=xG'(x)+G(x)-\frac{1}{1+x},$$

或

$$G'(x)+\left(\frac{1}{x}-\frac{1}{x^2}\right)G(x)=\frac{1}{x+x^2}-\frac{1}{x^2}. \tag{6.55}$$

等式(6.55)是线性一阶微分方程. 遗憾的是，它不容易求解.

可以证明，如果我们使用下面的指数生成函数取代普通生成函数的话，递推关系(6.53)式相当容易求解.

$$H(x)=\sum_{n=0}^{\infty}D_n\frac{x^n}{n!}.$$

为了确定 $H(x)$，我们用 $x^{n+1}/(n+1)!$ 乘以(6.53)式，并求和，得到

$$\sum_{n=0}^{\infty}D_{n+1}\frac{x^{n+1}}{(n+1)!}=\sum_{n=0}^{\infty}(n+1)D_n\frac{x^{n+1}}{(n+1)!}+\sum_{n=0}^{\infty}(-1)^{n+1}\frac{x^{n+1}}{(n+1)!}. \tag{6.56}$$

(6.56)式的左边是

$$H(x)-D_0=H(x)-1.$$

而右边第一项是

$$\sum_{n=0}^{\infty}D_n\frac{x^{n+1}}{n!}=x\sum_{n=0}^{\infty}D_n\frac{x^n}{n!}=xH(x).$$

右边第二项是

$$-\frac{x}{1!}+\frac{x^2}{2!}-\frac{x^3}{3!}+\cdots,$$

即 $e^{-x}-1$[参见(5.3)式]. 现在等式(6.56)变成

376

$$H(x)-1=xH(x)+e^{-x}-1.$$

因此，有

$$H(x)=\frac{e^{-x}}{1-x}.$$

我们可以把上式展开得到 D_n，它就是 $x^n/n!$ 的系数. 把 $H(x)$写成 $e^{-x}(1-x)^{-1}$，我们得到下面的表达式：

$$H(x)=\left[1-x+\frac{x^2}{2!}-\frac{x^3}{3!}+\cdots\right][1+x+x^2+\cdots]. \tag{6.57}$$

很容易直接看出

$$H(x)=\sum_{n=0}^{\infty}x^n\left[1-\frac{1}{1!}+\frac{1}{2!}-\frac{1}{3!}+\cdots+(-1)^n\frac{1}{n!}\right],$$

因此 $x^n/n!$ 的系数变成

$$D_n=n!\left[1-\frac{1}{1!}+\frac{1}{2!}-\frac{1}{3!}+\cdots+(-1)^n\frac{1}{n!}\right]. \tag{6.58}$$

等式(6.58)与我们早前的公式(6.14)一致.

从 $H(x)$推导出 D_n 的另一种方法是，通过观察可知，$H(x)$是序列(c_n)的普通生成函数，而(c_n)是序列$((-1)^n/n!)$和(1，1，1，…)的卷积. 因此，$H(x)$是$(n!c_n)$的指数生成函数. 从 $H(x)$推导出 D_n 的第三个方法将在练习 20 中加以说明.

我们以推导递推关系(6.53)式来结束本小节. 注意，根据(6.52)式，我们有

$$D_{n+1}-(n+1)D_n=D_{n+1}-nD_n-D_n=nD_{n-1}-D_n=-[D_n-nD_{n-1}].$$

因此，我们得出，对于所有 $j\geqslant1$，$k\geqslant1$，

$$(-1)^j[D_j - jD_{j-1}] = (-1)^k[D_k - kD_{k-1}].$$

现在有

$$(-1)^2[D_2 - 2D_1] = 1[1-0] = 1.$$

因此，对于所有的 $n \geqslant 0$，有

$$(-1)^{n+1}[D_{n+1} - (n+1)D_n] = (-1)^2[D_2 - 2D_1] = 1,$$

根据此式，对于所有的 $n \geqslant 0$，(6.53)式成立.

6.3.3 生成函数的联立方程

在6.1.4节，我们考虑了必须使用三个序列来确定满足递推关系方程组(6.19)式、(6.21)式和(6.23)式的情况，而不是一个序列. 生成函数的方法可以用于求解递推关系的方程组. 为了说明这一点，我们首先选择 a_0，b_0，c_0 使得(6.19)式、(6.21)式和(6.23)式成立. 使用 $a_1 = 2$，$b_1 = 1$，$c_1 = 1$，根据(6.19)式、(6.21)式和(6.23)式，我们有 377

$$2 = 2a_0 + b_0 + c_0,$$
$$1 = b_0 - c_0 + 1,$$
$$1 = c_0 - b_0 + 1.$$

这一方程组的一个解是取 $a_0 = 1$，$b_0 = c_0 = 0$. 使用这些值，我们可以假设(6.19)式、(6.21)式和(6.23)式对于 $k \geqslant 0$ 成立.

现在我们用 x^k 乘以上述每一个方程的两边，并对于 k 从0到 ∞ 求和. 我们得到方程组：

$$\sum_{k=0}^{\infty} a_{k+1}x^k = 2\sum_{k=0}^{\infty} a_k x^k + \sum_{k=0}^{\infty} b_k x^k + \sum_{k=0}^{\infty} c_k x^k,$$

$$\sum_{k=0}^{\infty} b_{k+1}x^k = \sum_{k=0}^{\infty} b_k x^k - \sum_{k=0}^{\infty} c_k x^k + \sum_{k=0}^{\infty} 4^k x^k,$$

$$\sum_{k=0}^{\infty} c_{k+1}x^k = \sum_{k=0}^{\infty} c_k x^k - \sum_{k=0}^{\infty} b_k x^k + \sum_{k=0}^{\infty} 4^k x^k.$$

如果方程

$$A(x) = \sum_{k=0}^{\infty} a_k x^k, \quad B(x) = \sum_{k=0}^{\infty} b_k x^k \quad 和 \quad C(x) = \sum_{k=0}^{\infty} c_k x^k$$

分别是序列 (a_k)，(b_k) 和 (c_k) 的普通生成函数，那么我们有

$$\frac{1}{x}[A(x) - a_0] = 2A(x) + B(x) + C(x),$$

$$\frac{1}{x}[B(x) - b_0] = B(x) - C(x) + \frac{1}{1-4x},$$

$$\frac{1}{x}[C(x) - c_0] = C(x) - B(x) + \frac{1}{1-4x}.$$

使用 $a_0 = 1$，$b_0 = c_0 = 0$，根据上述三个方程我们有

$$A(x) = \frac{1}{1-2x}[xB(x) + xC(x) + 1], \tag{6.59}$$

$$B(x) = \frac{1}{1-x}\left[-xC(x) + \frac{x}{1-4x}\right], \tag{6.60}$$

$$C(x) = \frac{1}{1-x}\left[-xB(x) + \frac{x}{1-4x}\right]. \tag{6.61}$$

从(6.60)式和(6.61)式容易看出

$$B(x) = C(x) = \frac{x}{1-4x}. \tag{6.62}$$

根据(6.59)式和(6.62)式，有

378

$$A(x)=\frac{2x^2-4x+1}{(1-2x)(1-4x)}. \tag{6.63}$$

使用(6.42)式，我们看到(6.62)式蕴涵着

$$B(x)=C(x)=\sum_{k=0}^{\infty}4^k x^{k+1}.$$

因此，对于 $k>0$，有 $b_k=c_k=4^{k-1}$，对于 $k=0$，有 $b_k=c_k=0$. 使用部分分数法展开(6.63)式的右边，我们得到

$$A(x)=\frac{1-3x}{1-4x}+\frac{x}{1-2x}.$$

上式可以重写成下面的表达式：

$$A(x)=1+\frac{x}{1-4x}+\frac{x}{1-2x}=1+\sum_{k=0}^{\infty}4^k x^{k+1}+\sum_{k=0}^{\infty}2^k x^{k+1}.$$

因此，对于 $k>0$，有 $a_k=4^{k-1}+2^{k-1}$，且 $a_0=1$. 这一结果容易检验. 特别地，我们有 $a_2=4+2=6$,这与我们在 6.1.4 节中所得到的结果一致.

本节练习

提醒读者注意：在下面每一道练习中，如果生成函数的分母原来有复根，那么作为答案给出生成函数就可以了.

1. 使用生成函数求解下列递推关系.

   (a)例 6.2 中的(6.5)式，初始条件是 $f(1)=0$.

   (b)例 6.3 中的(6.6)式.　　(c)例 6.3 中的(6.7)式.

2. 对于所给的初始条件，使用生成函数求解下列递推关系.

   (a)$a_{k+1}=a_k+3$，$a_0=1$　　(b)$a_{k+1}=3a_k+2$，$a_1=1$

   (c)$a_{k+2}=a_{k+1}-2a_k$，$a_0=0$，$a_1=1$

3. 使用生成函数求解 6.2 节练习 12 中的每一个递推关系.

4. 对于下列每一种情况，假设 $G(x)$是序列(a_k)的普通生成函数. 确定 a_k.

   (a)$G(x)=\dfrac{1}{(1-x)(1-3x)}$　　(b)$G(x)=\dfrac{2x+1}{(1-3x)(1-2x)}$

   (c)$G(x)=\dfrac{2x^2}{(1-3x)(1-5x)(1-7x)}$　　(d)$G(x)=\dfrac{1}{4x^2-5x+1}$

379

   (e)$G(x)=\dfrac{x}{x^2-5x+6}$　　(f)$G(x)=\dfrac{1}{8x^3-6x^2+x}$

5. 使用 6.3.3 节的结果，验证 6.1.4 节中我们得到的关于下列各项的值：

   (a)a_3　　(b)b_3　　(c)c_3

6. 考虑 6.1 节练习 18 的产品销售问题. 假设 $A(x)=\sum_{n=0}^{\infty}a_n x^n$是序列$(a_n)$的普通生成函数.

   (a)对于 $a_0=0$，$a_1=1$，确定 $A(x)$.　　(b)一般地，使用 a_0 和 a_1 求 $A(x)$.

   (c)对于 $a_0=a_1$，把你的答案运用于(b)证明 a_n 是常量.（根据这一递推关系，这是显然的.）

7. 对于 6.1 节练习 21，求 b_n.

8. 本练习要求读者利用生成函数的方法推导斐波那契数第 k 项的公式.

   (a)定义 F_{-1}很有用. 利用递推关系(6.34)式以及 F_1 和 F_0 的值推导出 F_{-1}的适当值.

   (b)如果 $G(x)=\sum_{k=0}^{\infty}F_k x^k$ 是 F_k 的普通生成函数，通过用 x^k 乘以递推关系(6.34)式并对 $k=0$ 到∞求和，推导出 $G(x)$的函数等式.

   (c)证明下式成立：

$$G(x)=\frac{1}{1-x-x^2}.$$

(d)确定 $1-x-x^2$ 的根，并利用这些根把 $1-x-x^2$ 写成 $(1-\lambda x)(1-\mu x)$ 的形式.

(e)利用部分分数法把

$$\frac{1}{(1-\lambda x)(1-\mu x)} \text{转换成} \frac{A}{1-\lambda x}+\frac{B}{1-\mu x}.$$

(f)根据(e)的结果，推导 F_k 的公式.

9. 假设对于 $n\geqslant 1$，G_n 满足下面的方程：

$$G_{n+1}=G_n+G_{n-1},$$

假设 $G_0=3$，$G_1=4$. 使用生成函数确定 G_n 的公式.

10. 使用生成函数求解例子 6.7 中的递推关系(6.10)式.

11. 求解下面的递推关系：

$$a_{k+1}=2a_k+k+5,\quad k\geqslant 0, a_0=0.$$

12. 假设对于 $n\geqslant 0$ 有 $a_{n+1}=(n+1)b_n$ 且 $a_0=0$. 如果 $A(x)$ 和 $B(x)$ 分别是 (a_n) 和 (b_n) 的普通生成函数，确定包含 $A(x)$，$B'(x)$ 和 $B(x)$ 的关系.

13. 假设对于 $k\geqslant 0$ 有

$$y_{k+1}=Ay_k+B,$$

其中 A 和 B 是实数，$A\neq 1$. 使用生成函数的方法确定以 y_0 表示的 y_k 的公式.

380

14. 假设对于 $k\geqslant 0$ 有

$$y_{k+2}-y_{k+1}+2y_k=4^k,$$

且假设有 $y_0=2$，$y_1=1$. 使用生成函数的方法确定 y_k.

15. 假设 Y_t 是时期 t 的国民收入. 根据 Samuelson[1939]，Goldberg[1958]推导出如下的递推关系：

$$Y_t=\alpha(1+\beta)Y_{t-1}-\alpha\beta Y_{t-2}+1,$$

其中 $t\geqslant 2$，α 和 β 是正常量. 假设 $Y_0=2$，$Y_1=3$，$\alpha=1/2$，$\beta=1$，确定序列 (Y_t) 的生成函数.

16. 对于 $\alpha=2$ 和 $\beta=4$，重复练习 15.

17. (Goldberg[1958])Metzler[1941]在他的关于存货周期的著作中，研究了为销售及为存货而进行生产的一位企业家在第 t 个时间周期的生产总收入 i_t. 他得到下面的递推关系：

$$i_{t+2}-2\beta i_{t+1}+\beta i_t=v_0,$$

其中 $t\geqslant 0$，且 β 是 $0<\beta<1$ 的常量，v_0 是正常量. 假设 $i_0=i_1=0$，确定序列 (i_t) 生成函数.

18. 如果对于 $n\geqslant 0$ 且 $C_0=1$，有

$$C_{n+1}=2nC_n+2C_n+2,$$

确定 C_n.

19. 求解 6.1 节练习 37 中推导出的递推关系.

20. 通过观察 $H(x)$ 是序列 (a_k) 和序列 (b_k) 的指数生成函数的积，其中 $a_k=(-1)^k$ 且 $b_k=k!$，根据 $H(x)$ 的方程(6.57)式推导出 D_n. 使用 5.5 节练习 18 的结果.

21. 如下推导 D_n 的公式.

(a)令

$$C_n=\frac{D_n}{n!}-\frac{D_{n-1}}{(n-1)!}.$$

确定以 C_n 表示的 C_{n+1} 的递推关系.

(b)通过迭代求解 C_n 的递推关系.

(c)使用 C_n 的公式求解 D_n.

22. 通过 6.3.3 节的方法求解 6.1 节练习 22 的递推关系.

23. 同时求解下列递推关系：

$$a_{n+1}=a_n+b_n+c_n,\quad n\geqslant 1,$$
$$b_{n+1}=4^n-c_n,\quad n\geqslant 1,$$
$$c_{n+1}=4^n-b_n,\quad n\geqslant 1,$$

满足初始条件 $a_1=b_1=c_1=1$.

24. (Anderson[1974])假设(a_n)满足下面的递推关系：

$$na_n=2(a_{n-1}+a_{n-2}),$$

381 其中 $n\geqslant 2$，$a_0=e$，$a_1=2e$. 设 $A(x)$是(a_n)的普通生成函数.

(a)证明 $A'(x)=2(1+x)A(x)$.

(b)确定 $A(x)$[提示：回想一下方程 $f'(x)=f(x)$].

25. 考虑常系数线性齐次递推关系(6.25)式. 本练习探讨使用特征根的求解方法与使用生成函数的求解方法之间的关系.

(a)证明序列(a_n)的普通生成函数 $G(x)$由 $G(x)=p(x)/q(x)$给出，其中

$$q(x)=1-c_1x-c_2x^2-\cdots-c_px^p$$

和

$$p(x)=a_0+(a_1-c_1a_0)x+(a_2-c_1a_1-c_2a_0)x^2+\cdots+(a_{p-1}-c_1a_{p-2}-\cdots-c_{p-1}a_0)x^{p-1}.$$

(b)证明：如果 α_1，α_2，$\cdots\alpha_p$ 是特征根，那么有

$$q(x)=(1-\alpha_1x)(1-\alpha_2x)\cdots(1-\alpha_px),$$

且 $q(x)$的根是 $1/\alpha_1$，$1/\alpha_2$，$\cdots 1/\alpha_p$.

(c)通过使用生成函数方法求解递推关系(6.29)式来说明这些结果，并与 6.2.1 节的结果进行比较.

(d)通过使用特征根方法求解递推关系(6.1)式来说明这些结果，并与 6.3.1 节的结果进行比较.

26. 使用方程(6.50)计算图 5.6 中的棋盘 B_J 的$R(x, B_J)$.

27. 使用 5.1 节练习 17 的结果、例 5.14 和方程(6.50)计算图 6.7 中的棋盘 B 的车多项式.

6.4 一些涉及卷积的递推关系[㊀]

6.4.1 简单有序根树的数量

在 3.5 节，我们已注意到凯莱把饱和碳氢化合物的计数问题归约成树的计数问题. 这里，我们讨论一个相关的问题，即计数简单有序根树(simple, ordered, rooted tree)，简称 SOR 树的数量问题. 这些是(无标签)根树[㊁]，在其每一个顶点有 0 个、1 个、或 2 个孩子的意义下是简单的. 同样，它们是有序的，每一个顶点的孩子都被标识为左(L)或右(R). 我们如下区分两棵 SOR 树：如果它们不是同构的，或者它们有不同的根，或者它们有相同根且同构，但是存在一个不同的左孩子或右孩子. 例如，图 6.8 中的两棵 SOR 树是不同的，即使它们同构且有相同的根. 382

我们设 u_n 是 n 个顶点的不同 SOR 树的数量. 图 6.9 表明，$u_1=1$，$u_2=2$，$u_3=5$. 习惯上把没有顶点的树当作一棵 SOR 树. 因此，我们有 $u_0=1$.

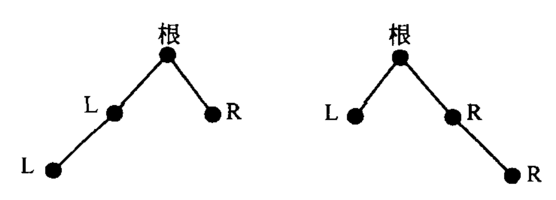


图 6.8 两棵不同的 SOR 树

假设 T 是一棵 $n+1$ 个顶点的 SOR 树，且 $n\geqslant 0$. 那么该树的根最多有两个孩子. 如果顶点 a 和 b 分别是一棵 SOR 树的根的左孩子和右孩子，那么

㊀ 本节的四个小节是相对独立的，原则上可以按任意顺序阅读. 从纯粹的教学法的观点看，如果没有足够的时间阅读全部四个小节，那么应该阅读 6.4.1 节、6.4.2 节、6.4.3 节中的一节，最好是阅读 6.4.1 节，然后阅读 6.4.4 节.

㊁ 对于根树的定义，参见 3.6.1 节.

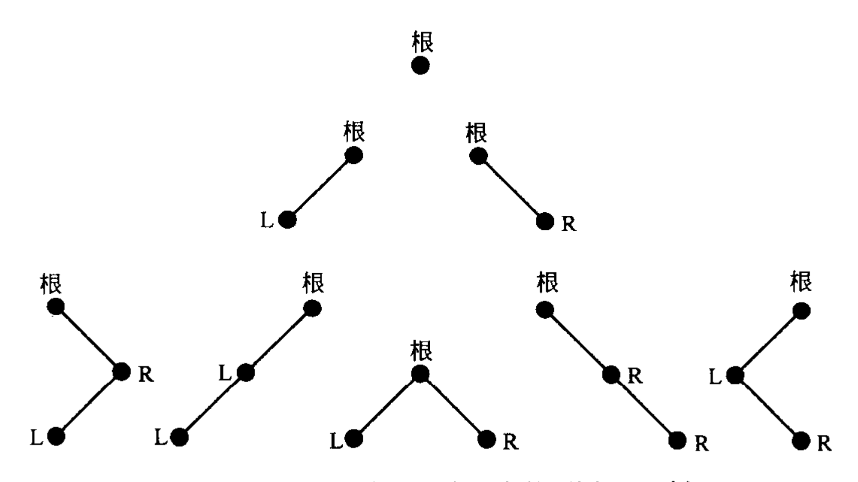


图 6.9 一个、两个和三个顶点的不同 SOR 树

顶点 a 和 b 自身分别是 SOR 树 T_L 和 T_R 的根(如果 a 或 b 不存在，那么对应的 SOR 树就是一棵没有顶点的树). 特别地，如果 T_R 有 r 个顶点，那么 T_L 有 $n-r$ 个顶点. 因此，我们有下面的递推关系：

$$u_{n+1}=u_0u_n+u_1u_{n-1}+u_2u_{n-2}+\cdots+u_nu_0, \tag{6.64}$$

其中 $n\geqslant 0$. 等式(6.64)给出已知前面的所有 u_i 的值($i\leqslant n$)时，计算 u_{n+1}的一个方法.

注意，(6.64)式的右边来自于一个卷积. 特别地，如果序列(v_n)定义为序列$(u_n)*(u_n)$，那么

$$u_{n+1}=v_n. \tag{6.65}$$

383

设 $U(x)=\sum_{n=0}^{\infty}u_nx^n$ 和 $V(x)=\sum_{n=0}^{\infty}v_nx^n$ 分别是序列(u_n)和(v_n)的普通生成函数. 那么根据(6.65)式，有

$$\sum_{n=0}^{\infty}u_{n+1}x^n=\sum_{n=0}^{\infty}v_nx^n.$$

因此

$$\frac{1}{x}[U(x)-u_0]=V(x),$$

所以有

$$\frac{1}{x}[U(x)-1]=V(x). \tag{6.66}$$

但是 $V(x)=U(x)U(x)=U^2(x)$，所以(6.66)式给出下面的等式：

$$xU^2(x)-U(x)+1=0. \tag{6.67}$$

等式(6.67)是 $U(x)$的函数等式. 我们可以通过把未知的 $U(x)$当成一个变量 y 来求解这一函数等式. 假设 $x\neq 0$，我们对下面的方程运用二次公式：

$$xy^2-y+1=0,$$

并求解 y，得到

$$y=U(x)=\frac{1\pm\sqrt{1-4x}}{2x}. \tag{6.68}$$

现在，我们可以通过展开求解 u_n. 特别地，我们注意到，可以使用二项式定理(定理 5.3)展开 $\sqrt{1-4x}$，给出下面的展开式：

$$(1-4x)^{1/2}=1+\frac{1}{2}(-4x)+\frac{\frac{1}{2}\left(-\frac{1}{2}\right)}{2!}(-4x)^2+\frac{\frac{1}{2}\left(-\frac{1}{2}\right)\left(-\frac{3}{2}\right)}{3!}(-4x)^3+\cdots$$

384

$$+\binom{\frac{1}{2}}{r}(-4x)^r+\cdots.$$

对于 $n\geqslant 1$，x^n 的系数可以写成下面的形式：

$$\binom{\frac{1}{2}}{n}(-4)^n=\frac{\left(\frac{1}{2}\right)\left(-\frac{1}{2}\right)\left(-\frac{3}{2}\right)\cdots\left(\frac{1}{2}-n+1\right)}{n!}(-4)^n$$

$$=\frac{\left(\frac{1}{2}\right)\left(-\frac{1}{2}\right)\left(-\frac{3}{2}\right)\cdots\left[-\frac{2n-3}{2}\right](-4)^n}{n!}$$

$$=\frac{\left(\frac{1}{2}\right)(-1)^{n-1}\left(\frac{1}{2}\right)\left(\frac{3}{2}\right)\cdots\left[\frac{2n-3}{2}\right](-1)^n4^n}{n!}$$

$$=\frac{\left(-\frac{1}{2^n}\right)[1\cdot 3\cdot 5\cdot\cdots\cdot(2n-3)]4^n}{n!}$$

$$=\frac{-2^n[1\cdot 3\cdot 5\cdot\cdots\cdot(2n-3)]}{n!}$$

$$=\frac{-2}{n}\,\frac{2^{n-1}}{(n-1)!}[1\cdot 3\cdot 5\cdot\cdots\cdot(2n-3)]$$

$$=\frac{-2}{n}\,\frac{2^{n-1}[1\cdot 3\cdot 5\cdot\cdots\cdot(2n-3)](n-1)!}{(n-1)!(n-1)!}$$

$$=\frac{-2}{n}\,\frac{[1\cdot 3\cdot 5\cdot\cdots\cdot(2n-3)][2\cdot 4\cdot 6\cdot\cdots\cdot(2n-2)]}{(n-1)!(n-1)!}$$

$$=\frac{-2}{n}\,\frac{(2n-2)!}{(n-1)!(n-1)!}=\frac{-2}{n}\binom{2n-2}{n-1}.$$

因此，有

$$(1-4x)^{1/2}=1-\sum_{n=1}^{\infty}\frac{2}{n}\binom{2n-2}{n-1}x^n. \tag{6.69}$$

现在，(6.68)式有两个符号，即有两个可能解. 如果我们取负号对(6.68)式求解，那么有

$$U(x)=\frac{1}{2x}[1-\sqrt{1-4x}],$$

所以用 $n+1$ 取代 n，有

$$U(x)=\frac{1}{2x}\left[\sum_{n=1}^{\infty}\frac{2}{n}\binom{2n-2}{n-1}x^n\right]=\sum_{n=1}^{\infty}\frac{1}{n}\binom{2n-2}{n-1}x^{n-1}$$

385

$$=\sum_{n=0}^{\infty}\frac{1}{n+1}\binom{2n}{n}x^n, \tag{6.70}$$

我们从(6.70)式可以得到下式：

$$u_n=\frac{1}{n+1}\binom{2n}{n}. \tag{6.71}$$

如果我们取正号对(6.68)式求解，那么我们发现，类似地有

$$U(x)=\frac{1}{x}-\sum_{n=1}^{\infty}\frac{1}{n}\binom{2n-2}{n-1}x^{n-1},\tag{6.72}$$

所以，对于$n\geqslant 1$，我们有

$$u_n=-\frac{1}{n+1}\binom{2n}{n}.\tag{6.73}$$

因为系数u_n必须是非负的(为什么)，所以(6.71)式一定是解，而(6.73)式不是. 我们也看到，为了得到$U(x)$，(6.68)式必须取负号. 我们也可以从(6.72)式直接看到这一点，因为如果$U(x)$是由(6.72)式给出的，那么$U(x)$将有一个项$1/x$，而$U(x)=\sum_{n=0}^{\infty}u_nx^n$.

由(6.71)式定义的数u_n称为卡塔兰数(Catalan number)，以 Eugene Charles Catalan 命名. 例如，我们发现有

$$u_0=\frac{1}{1}\binom{0}{0}=1,\qquad u_1=\frac{1}{2}\binom{2}{1}=1,\qquad u_2=\frac{1}{3}\binom{4}{2}=2,$$

$$u_3=\frac{1}{4}\binom{6}{3}=5,\qquad u_4=\frac{1}{5}\binom{8}{4}=14.$$

前四个结果与我们早前的计算相同，而第五项也容易检证. 我们将看到卡塔兰数在组合数学中是一个非常常用的数. 关于卡塔兰数在其他众多方面的应用可以参见 Eggleton 和 Guy[1988].

6.4.2 计算机中数的序列相乘的方法

假设给定n个数x_1，x_2，…，x_n的序列，而且我们希望求它们的积. 有很多求这个积的不同方法. 例如，假设$n=4$. 我们可以首先用x_1乘以x_2，然后把这个积再乘以x_3，再把这个积与x_4相乘. 另外，我们可以从x_1乘以x_2开始，然后，x_3乘以x_4，最后，把这两个积相乘. 我们可以通过在串$x_1x_2\cdots x_n$中适当地插入括号来区分这两种方法及其他方法[⊖]. 这时，第一种方法对应于下式：

386

$$(((x_1x_2)x_3)x_4),$$

而第二种方法对应于下式：

$$((x_1x_2)(x_3x_4)).$$

假设我们必须以给定的顺序进行乘法运算. 例如，我们不允许直接用x_3乘以x_1等. 假设给定n个数的一个序列. 那么有多少种方法指示计算机求这个序列的积呢？假设P_n表示所讨论的方法数量. 很容易看到，求这个积相当于以下面的方法在序列x_1，x_2，…，x_n中插入$n-1$个左括号和$n-1$个右括号：

1. 从不在单一项周围插入括号[即不允许(x_i)].

2. 当从左走到右时，右括号的数目从不超过左括号的数目.

注意$P_1=1$，因为只有一种方法插入0个左括号和0个右括号. 还有，$P_2=1$，$P_3=2$和$P_4=5$. 表6.4给出对应于这些数的括号. 很容易确定P_n的递推关系. 假设$n\geqslant 2$. 考虑最后进行的乘法. 这包括两个子积$x_1\cdots x_r$和

表6.4 进行两个、三个或四个数的乘法的方法

P_2	P_3	P_4
(x_1x_2)	$((x_1x_2)x_3)$	$(((x_1x_2)x_3)x_4)$
	$(x_1(x_2x_3))$	$(x_1(x_2(x_3x_4)))$
		$((x_1(x_2x_3))x_4)$
		$(x_1((x_2x_3)x_4))$
		$((x_1x_2)(x_3x_4))$

⊖ 括号不能区分是先进行x_1x_2然后再进行x_3x_4，然后求积的积，还是先进行x_3x_4，然后再进行x_1x_2，然后求积的积. 我们只考虑必须计算的积是什么.

$x_{r+1}\cdots x_n$ 的积. 即对于 $1<r<n-1$，我们有乘积

$$((x_1\cdots x_r)(x_{r+1}\cdots x_n)).$$

如果 $r=1$ 或 $n-1$，那么

$$(x_1(x_2\cdots x_n)) \quad 或 \quad ((x_1\cdots x_{n-1})x_n).$$

无论上面哪种情况，都有 P_r 种方法求第一个子积且有 P_{n-r} 种方法求第二个子积，所以对于 $n\geqslant 2$，我们得到下面的递推关系：

$$P_n=\sum_{r=1}^{n-1}P_rP_{n-r}, \tag{6.74}$$

现在，如果我们设 $P_0=0$，(6.74)式变成下面的形式：

$$P_n=\sum_{r=0}^{n}P_rP_{n-r}, \tag{6.75}$$

其中 $n\geqslant 2$. 现在设 $P(x)=\sum_{n=0}^{\infty}P_nx^n$ 是序列 (P_n) 的普通生成函数. 等式(6.75)表明 (P_n) 与卷积 $(P_n)*(P_n)$ 相关. 然而，因为(6.75)式只对于 $n\geqslant 2$ 成立，所以我们不能推断出 $P(x)=P^2(x)$. 为了克服这一困难，我们定义序列 (Q_n) 为序列 $(P_n)*(P_n)$. 那么注意

$$Q_n=\begin{cases}0=P_0P_0 & 如果\ n=0\\ 0=P_0P_1+P_1P_0 & 如果\ n=1\\ P_n & 如果\ n\geqslant 2.\end{cases} \tag{6.76}$$

387

而且，普通生成函数 $Q(x)=\sum_{n=0}^{\infty}Q_nx^n$ 满足下面的关系式：

$$Q(x)=P^2(x).$$

另外，通过(6.76)式，有

$$Q(x)=P(x)-x,$$

因为对于 $n\neq 1$，$P_n=Q_n$，而且 $P_1=1$，$Q_1=0$. 因此，我们有

$$P(x)-x=P^2(x).$$

这是 $P(x)$ 的函数等式. 我们通过把它重写成未知量 $y=P(x)$ 的二次方程

$$P^2(x)-P(x)+x=0,$$

并使用二次公式对其求解，得

$$P(x)=\frac{1\pm\sqrt{1-4x}}{2}. \tag{6.77}$$

为了求 P_n，我们可以使用二项式定理展开 $P(x)$. 另外，我们认识到，对于(6.68)式中的 $U(x)$，$P(x)$ 等于 $xU(x)$. 因此，对于 $n\geqslant 1$，$P_n=u_{n-1}$. 我们已定义 $P_0=0$. 根据公式(6.71)，对于 $n\geqslant 1$，有

$$P_n=\frac{1}{n}\binom{2n-2}{n-1}. \tag{6.78}$$

再一次出现卡塔兰数.

数 u_n 与 P_n 之间的密切关系表明，我们也许可以找到 SOR 树与乘法顺序之间的直接关系. 根据 Even[1973]，我们描述这样一种关系. 考虑有 n 个左括号和 n 个右括号的一个序列. 如果上面的条件 2 成立的话，这样的序列被称为*形式良好的*(well formed). 设 K_n 是这样的序列的数量. 那么显然有 $P_n=K_{n-1}$. 给定 n 个顶点的 SOR 树，把度为 1 的每一个顶点与括号()序列相对应. 把其他每一个顶点与下面的序列相对应：首先是“(”，后面是这个顶点的左孩子(如果存在的话)所对应的括号序列，后面跟着这个顶点的右孩子(如果存在的话)所对应的括号序列，后面跟

388

着“)”. 这样，每一个 SOR 树都有唯一的形式良好的括号顺列与其相对应，而这个序列被指定给这棵树的根. 图 6.10 说明了这一过程. 相反，给定一个有 n 个左括号和 n 个右括号的形式良好的括号序列，可以证明它来自于一棵 n 个顶点 SOR 树. 我们把这一证明留作练习(练习 17). 因此，$K_n=u_n$，而且我们再一次得到结论 $P_n=u_{n-1}$.

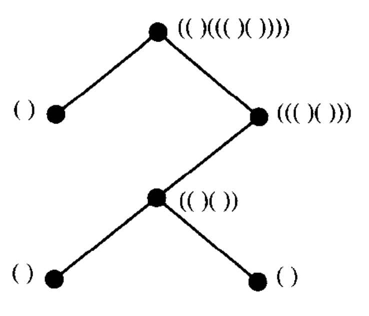


图 6.10 每一个顶点旁边是对应的形式良好的括号序列

6.4.3 RNA 中的二级结构

在 2.11 节和 2.12 节，我们研究了 RNA 分子中的基的线性链. 有时说这个链定义了 RNA 的*一级结构*(primary structure). 当 RNA 只在这个链中两个邻居之间有键时，则称这个链是*无规则卷曲*(random coil). 此时 RNA 不保留无规则卷曲. 它自己折叠起来形成称为 Watson-Crick 键的新键，创建螺线区域. 在 RNA 链 $s=s_1s_2\cdots s_n$ 的这样的 Watson-Crick 键合中，每一个基都可以至多键合到一个其他的非邻居的基上，而且如果 s_i 和 s_j 是键合的，且 $i<k<j$，那么 s_k 只能与位于 s_{i+1} 和 s_{j-1} 之间的基键合；即不存在交叉.[㊀] 这些新键定义原来 RNA 链的*二级结构*(secondary structure). 图 6.11 给出下面 RNA 链的一个可能的二级结构：

AACGGGCGGGACCCUUCAACCCUU.

Watson-Crick 键通常在基 A 和 U 或 G 和 C 之间形成，但是，根据 Howell，Smith 和 Waterman[1980]，我们发现在我们的讨论中允许所有可能键是很方便的. 在 RNA 链的研究中，Howell，Smith Waterman[1980]使用递推关系计算长度为 n 的 RNA 链的可能的二级结构的数量 R_n. 我们粗略地讨论他们的方法.[㊁]对于更多关于二级结构的研究，参见 Clote 和 Backofen[2000]、Setubal 和 Meidanis[1997]、Waterman[1995].

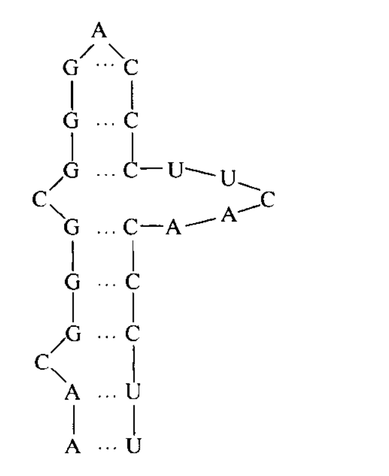


图 6.11 RNA 链 AACGGGCGGGACCCU-UCAACCCUU 的一个二级结构. Watson-Crick 键被表示成点线(在得到 SIAM 的许可之下从 Howell，Smith 和 Waterman[1980] 复制)

389

注意，因为不可能存在 Watson-Crick 键，所以 $R_1=R_2=1$. 还有，根据习惯设 $R_0=1$. 考虑长度为 $n+1\geqslant 3$ 的 RNA 链 $s_1s_2\cdots s_{n+1}$. 现在，或者 s_{n+1} 不形成 Watson-Crick 键合，或者它与 s_j 键合，$1\leqslant j\leqslant n-1$. 因此，对于 $n\geqslant 2$，

$$R_{n+1}=R_n+\sum_{j=1}^{n-1}R_{j-1}R_{n-j}, \qquad (6.79)$$

因为对于第一种情况，链 $s_1s_2\cdots s_n$ 可以自由地形成任意二级结构，而对于第二种情况，子链 $s_1s_2\cdots s_{j-1}$ 和 $s_{j+1}s_{j+2}\cdots s_n$ 自由地形成任意二级结构. 这是 R_n 的一个递推关系. 设 $R(x)$ 是 R_n 的普通生成函数，即

$$R(x)=\sum_{n=0}^{\infty}R_nx^n.$$

为了求 $R(x)$，我们注意到(6.79)式的右边的第二项几乎等于一个卷积. 如果我们定义 $(T_n)=(R_n)*(R_n)$，且我们使用 $R_0=1$ 的事实，那么对于 $n\geqslant 2$，我们有

㊀ 对于相关的键合问题，参见 Nussinov 等[1978].

㊁ 对于相关工作，参见 Stein 和 Waterman[1978]、Waterman[1978].

$$R_{n+1}=R_n+\sum_{j=1}^{n}R_{j-1}R_{n-j}-R_{n-1}R_0,$$

因此，对于 $n\geqslant 2$，有

390

$$R_{n+1}=R_n-R_{n-1}+T_{n-1}. \tag{6.80}$$

容易看到(6.80)式对于 $n=1$ 仍然成立. 进一步，如果我们定义 $R_{-1}=T_{-1}=0$，那么(6.80)式对于所有 $n\geqslant 0$ 成立. 现在，设 $T(x)=\sum_{n=0}^{\infty}T_nx^n$ 是 (T_n) 的普通生成函数. 根据(6.80)式，有

$$\sum_{n=0}^{\infty}R_{n+1}x^n=\sum_{n=0}^{\infty}R_nx^n-\sum_{n=0}^{\infty}R_{n-1}x^n+\sum_{n=0}^{\infty}T_{n-1}x^n.$$

因此，有

$$\frac{1}{x}[R(x)-R_0]=R(x)-x\left[\frac{R_{-1}}{x}+R(x)\right]+x\left[\frac{T_{-1}}{x}+T(x)\right],$$

所以有

$$\frac{1}{x}[R(x)-1]=R(x)-xR(x)+xT(x).$$

因为 (T_n) 是 (R_n) 与其自身的卷积，有

$$T(x)=R^2(x).$$

因此，我们有

$$\frac{1}{x}R(x)-\frac{1}{x}=R(x)-xR(x)+xR^2(x),$$

或者

$$x^2R^2(x)+(-x^2+x-1)R(x)+1=0.$$

对于 $x\neq 0$，我们有

$$R(x)=\frac{x^2-x+1\pm\sqrt{(-x^2+x-1)^2-4x^2}}{2x^2},$$

或者

$$R(x)=\frac{1}{2x^2}\left[x^2-x+1\pm\sqrt{1-(2x+x^2+2x^3-x^4)}\right]. \tag{6.81}$$

可以使用二项定理展开(6.81)式的平方根. 注意如果使用正号，那么存在项 $1/2x^2$，所以可以容易地确定在(6.81)式中是使用正号还是负号. 因此，负号一定是正确的. 通过考虑 x 趋近于 0 时的情况，我们也能看到这一点. 这时 $R(x)$ 应该趋近 $R(0)=R_0=1$. 如果我们在(6.81)式使用+号，当 x 趋近于 0 时，$R(x)$ 趋近于∞. 因此，负号一定是正确的.

6.4.4 由苯环所构建的有机化合物

Harary 和 Read[1970]以及 Anderson[1974]指出，由苯环所构建的某些有机化合物可以表示成六角形的形状，如图 6.12 所给出的例子. 计数不同种类的六角形形状是数学化学的一个中心课题.（关于这一课题的概述，可以参见 Cyvin Brunvoll 和 Cyvin[1992].）我们使用生成函数说明这一计数方法的思想.

391

我们对*多糖*(polyhese)感兴趣，这是一种六角形构形，在这一构形中所有六角形都是全等的正六角形，任意两个这样的六角形或者不相交，或者分享一个共同边，而且如果我们把它看成是一个图，那么这一构形是连通的. 我们增加一条限制：没有三个六角形能够相交于一点. 那么我们就得到*向下浓缩多糖*(catacondensed polyhese).

苯 萘 蒽 菲

图 6.12 由苯环所构建的某些有机化合物

同 Harary 和 Read[1970]一样，我们开始于一个向下浓缩多糖，且沿着这个多糖的周边在一条边上添加一个六角形且只考虑生成的向下浓缩多糖. 特别地，我们做如下简化假设：所有多糖都是由图 6.13 所示的带有一个基础边的基础六角形开始生成的. 在此基础上，我们只能把六角形附着于标签为 1，2 和 3 的边上. 因此，根据我们前面的假设，我们不能把六角形同时附着于边 1 和边 2 上或同时附着于边 2 和边 3 上. 我们称用这种方式构造出的总共有 n 个六角形的向下浓缩多糖为**多糖构形**(polyhexal configuration)，或简称为 **n 多糖**(n-polyhex). 一般地，一个 n 多糖(有 n 个六角形的多糖)可以通过在$(n-1)$多糖的一条边上附着一个新的六角形而得到. 这第 n 个六角形和这个$(n-1)$多糖只能有唯一一条共用边. 设 h_k 表示可能的 k 多糖的数量. 我们希望计算 h_k. 这里我们不直接推导 h_k 的递推关系，而是引入其他两个序列，并利用这两个序列计算 h_k. 设 s_k 表示这样的构形数量：只有一个六角形与基础六角形连接，而 d_k 表示这样的数量：正好有两个六角形与基础六角形连接. 显然，有

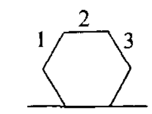


图 6.13 一个基础六角形

$$h_k = s_k + d_k, \tag{6.82}$$

其中 $k \geqslant 2$. 然而，(6.82)式对于 $k=1$ 不成立，因为 $h_1=1$，$s_1=d_1=0$. 如果我们有一个$(k+1)$多糖且只有一个六角形与基础六角形相连接，那么有 3 种可能的连接边，所以有

$$s_{k+1} = 3h_k, \tag{6.83}$$

其中 $k \geqslant 1$. 如果我们有一个$(k+1)$多糖且有两个六角形与基础六角形连接，那么它们必定使用图 6.13 的边 1 和边 3，因为三个六角形不能交相于一点. 因此，一个 r 多糖与边 1 连接，而一个 $k-r$ 多糖与边 3 连接，其中 $1 \leqslant r \leqslant k-1$. 我们得出，对于 $k \geqslant 2$，有

$$d_{k+1} = h_1 h_{k-1} + h_2 h_{k-2} + \cdots + h_{k-1} h_1 \tag{6.84}$$

392

注意，(6.82)式、(6.83)式和(6.84)式三个递推关系可以一起用于迭代计算所需的数 h_k. 已知 h_1，h_2，…，h_k，我们使用(6.83)式和(6.84)式分别计算 s_{k+1} 和 d_{k+1}，然后根据(6.82)式得到 h_{k+1}. 这种情况类似于 6.1.4 节的情况，在那里我们首次遇到了递推关系的联立方程.

现在考虑普通生成函数：

$$H(x) = \sum_{k=0}^{\infty} h_k x^k, \quad S(x) = \sum_{k=0}^{\infty} s_k x^k, \quad D(x) = \sum_{k=0}^{\infty} d_k x^k. \tag{6.85}$$

我们要计算 $H(x)$. 确定这一生成函数的技术与前面各小节的技术多少有点不同. 那里，在运算生成函数时，要利用 5.2 节的结果. 注意，h_k，s_k，d_k 都只对 $k=1$ 到∞有定义. 为了使用(6.85)式的生成函数，便利的做法是定义 $h_0=s_0=d_0=0$，因此可以从 0 到∞求和. 如果我们取 $h_0=s_0=d_0=0$，那么(6.82)式对 $k=0$ 以及 $k \geqslant 2$ 成立. 又因为 $s_1=0$ 和 $h_0=0$，所以(6.83)式对所有 $k \geqslant 0$ 成立. 因为 $h_0=0$，所以我们可以把 $h_0 h_k + h_k h_0$ 加到(6.84)式的右边，于是得到，对于所有 $k \geqslant 2$，有

$$d_{k+1} = h_0 h_k + h_1 h_{k-1} + h_2 h_{k-2} + \cdots + h_k h_0, \tag{6.86}$$

但是很容易看到对于所有 $k \geqslant 0$，(6.86)式成立，因为 $d_1=0=h_0 h_0$，$d_2=0=h_0 h_1 + h_1 h_0$.

使用(6.82)式，我们尝试通过 5.2 节的方法得出结论：

$$H(x) = S(x) + D(x).$$

然而，上式不成立，因为(6.82)式对于 $k=1$ 不成立．如果我们定义下式：

$$g_k = \begin{cases} h_k & 如果 k \neq 1 \\ 0 & 如果 k = 1, \end{cases}$$

那么，对于所有 $k\geqslant 0$，有

$$g_k = s_k + d_k$$

成立．另外，如果

393

$$G(x) = \sum_{k=0}^{\infty} g_k x^k,$$

那么

$$G(x) = S(x) + D(x).$$

最后

$$H(x) = G(x) + x. \tag{6.87}$$

因为序列(h_k)是序列(g_k)与序列(0，1，0，…)的和．因此，有

$$H(x) = S(x) + D(x) + x. \tag{6.88}$$

接下来，我们已看到，对于 $k\geqslant 0$，(6.83)式成立．因此，有

$$\sum_{k=0}^{\infty} s_{k+1} x^k = 3\sum_{k=0}^{\infty} h_k x^k,$$

或

$$\frac{1}{x}[S(x) - s_0] = 3H(x),$$

而且因为 $s_0=0$，所以有

$$\frac{1}{x}S(x) = 3H(x).$$

因此，有

$$S(x) = 3xH(x). \tag{6.89}$$

最后，我们来化简 $D(x)$．对于 $k\geqslant 0$，令 $e_k=d_{k+1}$，我们从(6.86)式看到，(e_k)是序列(h_k)与其自身的卷积．令

$$E(x) = \sum_{k=0}^{\infty} e_k x^k,$$

我们有

$$E(x) = H^2(x). \tag{6.90}$$

那么，

$$D(x) = \sum_{k=0}^{\infty} d_k x^k = \sum_{k=1}^{\infty} d_k x^k = \sum_{k=1}^{\infty} e_{k-1} x^k = x\sum_{k=1}^{\infty} e_{k-1} x^{k-1} = x\sum_{k=0}^{\infty} e_k x^k = xE(x).$$

因此，根据(6.90)式，我们有

$$D(x) = xH^2(x). \tag{6.91}$$

在(6.88)式中使用(6.89)式和(6.91)式给出下式：

394

$$H(x) = 3xH(x) + xH^2(x) + x,$$

或者

$$xH^2(x) + (3x-1)H(x) + x = 0. \tag{6.92}$$

方程(6.92)是未知函数 $H(x)$的二次方程．我们用二次公式求解这一方程，得到

$$H(x)=\frac{1}{2x}[1-3x\pm\sqrt{(3x-1)^2-4x^2}],$$

或者：

$$H(x)=\frac{1}{2x}[1-3x\pm\sqrt{1-(6x-5x^2)}],\tag{6.93}$$

其中 $x\neq0$. 可以使用二项式定理展开 $H(x)$，一旦得到展开式，就可以选择适当的符号，正号或者负号.

遗憾的是，刚才我们所描述的计算 h_k 的方法是有缺陷的. 有可能在使用我们描述的这一方法构建 k 多糖时，以有三个六角形相交于一点的结构而告终，即违反了向下浓缩的条件.（这一事实的证明留给读者作为练习，参见练习 23.）这一方法的第二个麻烦是，它可能给出下面这样的构形：在一个环内环绕并返回它们自己，而在这个环内部的某个地方封闭了一个不是这个构形的一部分的六角形. 还可能不是沿着边结束，而是以重叠的六角形结束.（这两个附加的麻烦的证明也留给读者作为练习，参见练习 23.）因此，尽管我们讨论的这一计数方法很巧妙，但是它过高估计了 k 多糖的数量，而且它计数了不理想的诸如环的构形. Harary 和 Read 探讨了我们讨论的这类问题. 他们使违背没有三个六角形相交于一点的性质的情况，和违背六角形只沿着公共边相交的性质的情况合法化，方法是把这个构形分成若干层次，且体系化在某点从一个层次到达另一个层次. 这样，一旦一个六角形构形环绕并返回到其自身形成一个环，或三个六角形相交于一点，我们就认为这一构形到了另一个层次，它位于这一构形的前面部分的上面并于其交迭. 这就是这一类文献中的标准思想(参见 Cyvin，Brunvoll 和 Cyvin[1992]).

我们所讨论的困难说明了这样的事实：利用生成函数或其他方法得到六角形构形的精确计数还没有取得完全成功. 在很大程度上，在计数各种构形方面，计算机生成方法取代了生成函数. Knop 等[1983]首先发表了 $k\leqslant10$ 时的 k 多糖的结果. 更进一步的结果还有待于完成，例如，请参见 Toosic 等[1995].

本节练习

1. 验证卡塔兰数 $u_4=14$ 的确计数四个顶点的 SOR 树的数量.
2. 计算卡塔兰数 u_5 和 u_6. [395]
3. 使用(6.78)式计算 P_5 并检验它的确计数使 5 个数的序列相乘的方法数量.
4. 根据递推关系(6.79)式和相应的初始条件计算 R_3，R_4 和 R_5. 通过画出相应的二级结构加以验证.
5. 根据递推关系(6.82)～(6.84)式和相应的初始条件计算 h_2，h_3和 h_4，并通过画出相应的多糖验证你的答案.
6. 在生成 n 多糖时，如果我们取消第 n 个六角形与$(n-1)$多糖只有唯一一条共用边的假设，那么在多糖中可以形成环及内部的"洞". 如果正好只有一个洞，而且它的大小是一个六角形，那么这个多糖称为*环烯*(circulene).
   (a)形成一个环烯最少需要多少个六角形?
   (b)形成有一个洞且它的大小至少是两个六角形的环烯最少需要多少个六角形(这样的多糖称为*冠状体*(coronoid))?
   (c)形成有两个洞且每一个洞的大小至少是两个六角形的冠状体最少需要多少个六角形?
7. 一棵*根树*(rooted tree)称为是*有序的*(ordered)，如果对给定的顶点的所有孩子指定一个从左到右的固定顺序. 另一种方法是，用整数 1，2，…，k 给给定顶点的 k 个孩子赋标签. 两个有序根树被认为是不同的，如果它们不同构，或者有一个不同的根，或者它们是同构的、有相同的根，但是两个相关顶点的孩子的顺序是不同的. 假设 r_n 是 n 个顶点有序根树的数量，而 $r_n(k)$是 n 个顶点的，根的度为 k 的有序根树的数量.
   (a)利用 $r_n(k)$确定 r_n 的表达式.
   (b)利用其他 r_n 确定 $r_n(2)$的表达式.

8. 假设 $A(x)=\sum_{n=0}^{\infty}a_nx^n$, $B(x)=\sum_{n=0}^{\infty}b_nx^n$ 和 $C(x)=\sum_{n=0}^{\infty}c_nx^n$ 分别是序列(a_n)，(b_n)和(c_n)的普通生成函数. 假设 $a_0=b_0=c_0=0$，$a_1=c_1=0$，$b_1=1$，$a_2=b_2=0$，$c_2=1$.
   (a)假设对于 $n\geqslant3$，$c_n=a_n+b_n$. 把这一表达式转换成使用生成函数的陈述.
   (b)假设对于 $n\geqslant0$，有 $a_{n+1}=4c_n$. 把这一表达式转换成使用生成函数的陈述.
   (c)假设 $n\geqslant2$ 有 $b_{n+1}=c_1c_{n-1}+c_2c_{n-2}+\cdots+c_{n-1}c_1$. 把这一表达式转换成使用生成函数的陈述.
   (d)使用(a)～(c)的答案推导只包含 $C(x)$的方程.
9. 假设 $A(x)$，$B(x)$，$C(x)$，a_0，b_0，c_0，a_1，b_1，c_1，a_2 和 b_2 都如练习 8 所示的那样，且 $c_2=4$.
   396
   (a)假设对于 $n\geqslant3$，有 $c_n=a_n+2b_n+2$. 把这一表达式转换成使用生成函数的陈述.
   (b)假设对于 $n\geqslant0$，有 $a_{n+1}=3c_n$. 把这一表达式转换成使用生成函数的陈述.
   (c)假设对于 $n\geqslant2$，有 $b_{n+1}=c_1c_{n-1}+c_2c_{n-2}+\cdots+c_{n-1}c_1$. 把这一表达式转换成使用生成函数的陈述.
   (d)使用(a)～(c)的答案推导只包含 $C(x)$的方程.
10. 如果 $H(x)$是由(6.93)式给出的，你如何区分在计算 $H(x)$中是使用正号还是使用负号?
11. (a)使用 $H(x)$的公式[方程(6.93)]计算数 h_1.
   (b)对 h_2 重复(a).　　　　(c)对 h_3 重复(a).
12. 通过确定两个差为 u_n 的二项式系数证明，对于$n=0$，1，2，…，卡塔兰数 $u_n=\frac{1}{n+1}\binom{2n}{n}$是整数. $\Big[$提示：考虑$\binom{2n}{n}$. $\Big]$
13. (Waterman[1978])此练习确定 R_n 的下界，其中 R_n 是长度为 n 的 RNA 链的可能二级结构的数量.
   (a)根据方程(6.79)和初始条件，证明对于 $n\geqslant2$，有
$$R_{n+1}=R_n+R_{n-1}+\sum_{k=1}^{n-2}R_kR_{n-k-1}. \tag{6.94}$$
   (b)通过使用方程(6.94)及用 n 取代(6.94)式中的 $n+1$ 所得到的方程，根据初始条件证明对于 $n\geqslant2$，有
$$R_{n+1}=R_n+R_{n-1}+R_{n-2}+\sum_{k=1}^{n-3}R_kR_{n-k-1}. \tag{6.95}$$
   (c)因为 $R_{p+1}\geqslant R_p$，得出结论：对于 $n\geqslant2$，$R_{n+1}\geqslant2R_n$，所以，对于 $n\geqslant2$，有 $R_n\geqslant2^{n-2}$.
14. (Riordan[1975])假设 $2n$ 个点分布于一个圆的圆周上. 把这些点配对，并通过这个圆的弦连接相应的点. 证明做这一配对并使得没有弦交叉的方法数量由卡塔兰数给出(Maurer[1992]和 Nussinov 等[1978]把这种弦的交叉模式与生物化学问题联系起来，而 Ko[1979]、Read[1979]和 Riordan[1975]研究了得到弦正好交叉 k 次的方法数量问题).
15. (Even[1973])后进先出栈(Last In-First Out(LIFO)stack)是类似于自助餐厅中叠罗的盘子的一种记忆设备：放在这一叠盘子最顶端的最后一个盘子是可以被拿走的第一个盘子. 标签为 1，2，…，n 的项的序列正等待着被放入一个空的 LIFO 栈中. 这一序列必须以给定的顺序入栈. 然而，我们可以随时移出一项. 移出的项不再回到栈中或等待存储的序列中. 最终，我们从栈中移出所有项，并得到标签 1，2，…，n 的一个排列. 例如，如果 $n=3$，我们可以先移入 1，然后移入 2，然后移出 2，移出 1，然后移入 3，最后，移出 3，得到一个排列 2，1，3. 设 q_n 是能够这样得到的排列的数量.
   397
   (a)确定 q_1，q_2，q_3 和 q_4.
   (b)通过得到一个递推关系并对其求解来确定 q_n.
16. 对于 6.4.1 节，假设我们用 $h_0=-3/2$ 取代 $h_0=0$[⊖].
   (a)证明对于所有 $k\geqslant2$，有

⊖ 此想法来自于 Martin Farber[私人通信].

$$h_{k+1}=h_0h_k+h_1h_{k-1}+\cdots+h_kh_0.$$

(b)定义(c_k)为序列$(h_k)*(h_k)$且设$w_k=h_{k+1}$. 令$C(x)$为(c_k)的普通生成函数，且$W(x)$为(w_k)的普通生成函数. 求$W(x)$和$C(x)$与$H(x)$的关系并推导出$H(x)$的函数等式.

(c)求解$H(x)$.

(d)为什么(c)的答案不同于(6.93)式给出的$H(x)$的公式？新的$H(x)$与旧的$H(x)$的关系是什么？

17. 通过6.4.2节给出的方法证明每一个形式良好的n个左括号和n个右括号的序列都得自于某棵SOR树.

18. 设v_n计数在一次竞选中n张投票投给两个候选人A和B的每一个候选人的方法数量，其中A从不输给B. 通过给出n张投票的顺序与n个数的乘积的顺序之间的直接关系确定v_n.

19. (Anderson[1974])简单偏序根树(simple, partly ordered, rooted tree(SPR tree))是这样的简单根树，其中只有当一个顶点有两个孩子时，才把标签L和R赋给该顶点的两个孩子. 图6.14给出若干SPR树. 设u_n计数n个顶点的SPR树的数量，a_n计数根有一个孩子的n个顶点的SPR树的数量，且b_n计数根有两个孩子的n个顶点的SPR树的数量. 假设$a_0=b_0=u_0=0$. 令$U(x)$、$A(x)$和$B(x)$分别是(u_n)、(a_n)和(b_n)的普通生成函数.

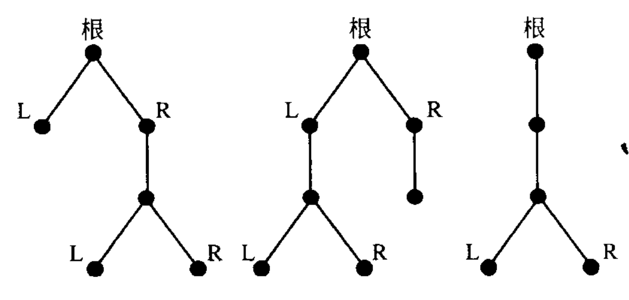


图6.14 若干SPR树

(a)计算a_1，b_1，u_1，a_2，b_2和u_2. 398

(b)推导出使用a_n和b_n给出u_n且对所有$n\neq 1$都成立的关系.

(c)推导出使用u_n给出a_{n+1}且对所有$n\geqslant 0$都成立的关系.

(d)推导出使用u_1，u_2，…，u_n给出b_{n+1}且对所有$n\geqslant 2$都成立的关系.

(e)推导出使用u_0，u_1，u_2，…，u_n给出b_{n+1}且对所有$n\geqslant 0$都成立的关系.

(f)根据(b)、(c)和(e)的答案确定u_3，u_4，a_3，a_4，b_3和b_4，并通过画出SPR树来检验答案.

(g)把(b)的答案转换成使用生成函数的陈述.

(h)把(c)的答案转换成使用生成函数的陈述.

(i)把(e)的答案转换成使用生成函数的陈述.

(j)证明下式成立：

$$U(x)=\frac{1}{2x}[1-x\pm\sqrt{(x-1)^2-4x^2}].$$

20. (Liu[1968])假设$A(x)$是序列(a_n)的普通生成函数，而$B(x)$是序列(b_n)的普通生成函数，且假设

$$b_n=a_{n-1}b_0+a_{n-2}b_1+\cdots+a_0b_{n-1},n\geqslant 1.$$

确定$A(x)$和$B(x)$的关系.

21. 对于下面的情况扩展练习20的结果：

$$b_n=a_{n-r}b_0+a_{n-r-1}b_1+\cdots+a_0b_{n-r}$$

对于$n\geqslant k$成立，其中$k\geqslant r$.

22. (Liu[1968])回想6.1节练习37中引入的位串中的模式的定义. 设a_n是下面的n位数字位串的数量：在这些位串中，模式010首次出现在第n个数字处.

(a)证明对于$n\geqslant 6$，有$2^{n-3}=a_n+a_{n-2}+a_{n-3}2^0+a_{n-4}2^1+\cdots+a_3 2^{n-6}$.

(b)设$b_0=1$，$b_1=0$，$b_2=1$，$b_3=2^0$，$b_4=2^1$，$b_5=2^2$，…，$a_0=a_1=a_2=0$，$n\geqslant 3$，有

$$2^{n-3}=a_nb_0+a_{n-1}b_1+a_{n-2}b_2+\cdots+a_0b_n.$$

(c)设$A(x)$和$B(x)$分别是序列(a_n)和(b_n)的普通生成函数，把(b)中得到的方程转换成包含$A(x)$和$B(x)$的陈述(见练习21).

(d)求解$A(x)$.

23. 证明：使用我们在6.4.4节中所描述的构建k多糖的方法有可能结束于下面的构形： 399

(a)存在相交于一点的三个六角形，即违反了向下浓缩的条件.

(b)沿着一个圆环绕并回到自身，在这个圆内部某处封闭一个不是这一构形一部分的六角形.

(c)存在重叠的而不是沿着边相交的六角形.

参考文献

ADBY, P. R., and DEMPSTER, M. A. H., *Introduction to Optimization Methods*, Chapman & Hall, London, 1974.

ADLER, I., "The Consequence of Constant Pressure in Phyllotaxis," *J. Theor. Biol., 65* (1977), 29–77.

ANDERSON, I., *A First Course in Combinatorial Mathematics*, Clarendon Press, Oxford, 1974.

APOSTOLICO, A., and GIANCARLO, R., "Sequence Alignment in Molecular Biology," in M. Farach-Colton, F. S. Roberts, M. Vingron, and M. S. Waterman (eds.), *Mathematical Support for Molecular Biology*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 47, American Mathematical Society, Providence, RI, 1999, 85–115.

BATSCHELET, E., *Introduction to Mathematics for Life Scientists*, Springer-Verlag, New York, 1971.

BRUALDI, R. A., *Introductory Combinatorics*, 3rd ed., Prentice Hall, Upper Saddle River, NJ, 1999.

CLOTE, P., and BACKOFEN, R., *Computational Molecular Biology: An Introduction*, Wiley, New York, 2000.

COXETER, H. S. M., *Introduction to Geometry*, Wiley, New York, 1969.

CYVIN, B. N., BRUNVOLL, J., and CYVIN, S. J., "Enumeration of Benzenoid Systems and Other Polyhexes," in I. Gutman (ed.), *Advances in the Theory of Benzenoid Hydrocarbons II*, Springer-Verlag, Berlin, 1992, 65–180.

EGGLETON, R. B., and GUY, R. K., "Catalan Strikes Again! How Likely Is a Function to Be Convex?," *Math. Mag., 61* (1988), 211–219.

ELAYDI, S. N., *An Introduction to Difference Equations*, Springer-Verlag, New York, 1999.

EVEN, S., *Algorithmic Combinatorics*, Macmillan, New York, 1973.

FOWLER, D. R., PRUSINKIEWICZ, P., and BATTJES, J., "A Collision-Based Model of Spiral Phyllotaxis," *Computer Graphics, 26* (1992), 361–368.

FU, Z. L., "The Number of Latin Rectangles," *Math. Practice Theory, 2* (1992), 40–41.

GAMOW, G., *One, Two, Three . . . Infinity*, Mentor Books, New American Library, New York, 1954.

GOLDBERG, S., *Introduction to Difference Equations*, Wiley, New York, 1958.

GUSFIELD, D., *Algorithms on Strings, Trees and Sequences; Computer Science and Computational Biology*, Cambridge University Press, New York, 1997.

HARARY, F., and PALMER, E. M., *Graphical Enumeration*, Academic Press, New York, 1973.

HARARY, F., and READ, R. C., "The Enumeration of Tree-like Polyhexes," *Proc. Edinb. Math. Soc., 17* (1970), 1–14.

HOLLINGDALE, S. H., "Methods of Operational Analysis," in J. Lighthill (ed.), *Newer Uses of Mathematics*, Penguin Books, Hammondsworth, Middlesex, England, 1978, 176–280.

HOWELL, J. A., SMITH, T. F., and WATERMAN, M. S., "Computation of Generating Functions for Biological Molecules," *SIAM J. Appl. Math., 39* (1980), 119–133.

KELLEY, W. G., and PETERSON, A. C., *Difference Equations. An Introduction with Applications*, Harcourt/Academic Press, San Diego, CA, 2001.

KIEFER, J., "Sequential Minimax Search for a Maximum," *Proc. Amer. Math. Soc., 4* (1953), 502–506.

400

KNOP, J. V., SZYMANSKI, K., JERIČEVIĆ, O., and TRINAJSTIĆ, N., "Computer Enumeration and Generation of Benzenoid Hydrocarbons and Identification of Bay Regions," *J. Comput. Chem.*, *4* (1983), 23–32.

KO, C. S., "Broadcasting, Graph Homomorphisms, and Chord Intersections," Ph.D. thesis, Department of Mathematics, Rutgers University, New Brunswick, NJ, 1979.

LIU, C. L., *Introduction to Combinatorial Mathematics*, McGraw-Hill, New York, 1968.

MACMAHON, P. A., "A New Method and Combinatory Analysis with Application to Latin Squares and Associated Questions," *Trans. Cambridge Philos. Soc.*, *16* (1898), 262–290.

MARKOWSKY, G., "Misconceptions about the Golden Ratio," *The College Mathematics Journal*, *23* (1992), 2–19.

MAURER, S. B., "A Minimum Cycle Problem in Bacterial DNA Research," RUTCOR (Rutgers Center for Operations Research) Research Report 27-92, 1992.

METZLER, L., "The Nature and Stability of Inventory Cycles," *Rev. Econ. Statist.*, *23* (1941), 113–129.

MYERS, E. W., "Seeing Conserved Signals: Using Algorithms to Detect Similarities Between Biosequences," in E. S. Lander and M. S. Waterman (eds.), *Calculating the Secrets of Life*, National Academy Press, Washington, DC 1995, 56–89.

NUSSINOV, R. P., PIECZENIK, G., GRIGGS, J. R., and KLEITMAN, D. J., "Algorithms for Loop Matchings," *SIAM J. Appl. Math.*, *35* (1978), 68–82.

READ, R. C., "The Chord Intersection Problem," *Ann. N.Y. Acad. Sci.*, *319* (1979), 444–454.

RIORDAN, J., "The Distribution of Crossings of Chords Joining Pairs of $2n$ Points on a Circle," *Math. Comp.*, *29* (1975), 215–222.

RYSER, H. J., *Combinatorial Mathematics*, Carus Mathematical Monographs No. 14, Mathematical Association of America, Washington, DC, 1963.

SAMUELSON, P. A., "Interactions between the Multiplier Analysis and the Principle of Acceleration," *Rev. Econ. Statist.*, *21* (1939), 75–78. (Reprinted in *Readings in Business Cycle Theory*, Blakiston, Philadelphia, 1944.)

SCHIPS, M., *Mathematik und Biologie*, Teubner, Leipzig, 1922.

SETUBAL, J. C., and MEIDANIS, J., *Introduction to Computational Molecular Biology*, PWS Publishers, Boston, 1997.

SHANNON, C. E., "The Zero-Error Capacity of a Noisy Channel," *IRE Trans. Inf. Theory*, *IT-2* (1956), 8–19.

SHAO, J. Y., and WEI, W. D., "A Formula for the Number of Latin Squares," *Discrete Math.*, *110* (1992), 293–296.

SLOANE, N. J. A. (ed.), *The On-Line Encyclopedia of Integer Sequences*, published electronically at http://www.research.att.com/~njas/sequences/ (2003).

STEIN, P. R., and WATERMAN, M. S., "On Some New Sequences Generalizing the Catalan and Motzkin Numbers," *Discrete Math.*, *26* (1978), 261–272.

401

TAKÁCS, L., "The Problem of Coincidences," *Arch. Hist. Exact Sci.*, *21* (1980), 229–244.

TOŠIĆ, R., MASULOVIC, D., STOJMENOVIC, I., BRUNVOLL, J., CYVIN, B. N., and CYVIN, S. J., "Enumeration of Polyhex Hydrocarbons to $h = 17$," *J. Chem. Inf. Comput. Sci.*, *35* (1995), 181–187.

WATERMAN, M. S., "Secondary Structure of Single-Stranded Nucleic Acids," *Studies on Foundations and Combinatorics, Advances in Mathematics Supplementary Studies*, Vol. 1, Academic Press, New York, 1978, 167–212.

WATERMAN, M. S., *Introduction to Computational Biology; Maps, Sequences and Genomes*, CRC Press, Boca Raton, FL, 1995.

402

第7章 容斥原理

7.1 容斥原理及其应用

7.1.1 一些简单例子

本章介绍另一个基本的计数工具，这就是容斥原理. 我们用下面的例子介绍这一原理.

【例 7.1 求职者】 假设有 18 名求职者组成一个小组，10 个人有计算机编程技术，5 个人有统计技术，2 个人既有编程技术又有统计技术. 这一组人中有多少人什么技术都没有？为了回答这一问题，我们画出一个维恩图，如图 7.1 所示[⊖]. 总共有 18 个人. 为了弄清楚有多少人什么技术都没有，我们要从 18 个中减去有编程技术的 10 个人，再减去有统计技术的 5 个人. 然而，我们可能把其中的几个人减去了两次. 即所有具有两种技术的人(这一数目就是图 7.1 中的有编程技术的人的集合与有统计技术的人的集合的交集中的人的数目.)都被减去了两次. 本例中有两个这样的人. 因此，我们必须把这两个人再加回去以得到正确的计数. 总之，我们得到

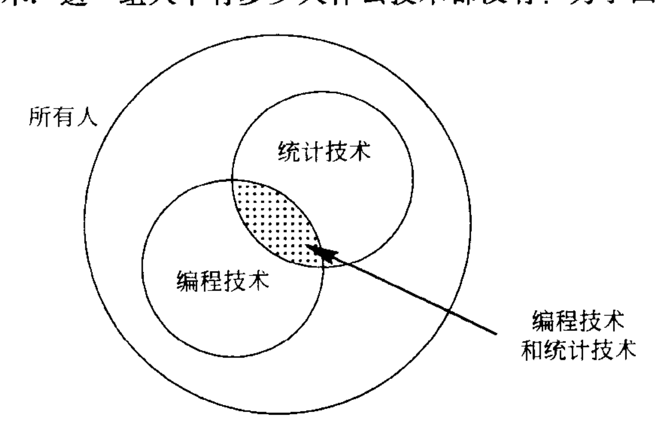


图 7.1　例 7.1 的维恩图

$$18-10-5+2=5$$

403 是什么技术都没有的人的数目. 我们将扩展上述推理. ■

假设我们有 N 个对象的集合 A. 设 a_1，a_2，…，a_r 是这些对象可能有的性质的集合，设 $N(a_i)$是有性质 a_i 的对象数目. 一个对象可能有若干所讨论的性质(或一个性质也没有). 设 $N(a_i')$计数没有性质 a_i 的对象数目. 这时，我们有

$$N=N(a_i)+N(a'_i).$$

因为一个对象可能有多个性质，所以计数既有性质 a_i 又有性质 a_j 的对象数目很有用. 这一数目记为 $N(a_ia_j)$. 既没有性质 a_i 又没有性质 a_j 的对象数目记为 $N(a_i'a_j')$，有性质 a_j 但没有性质 a_i 的对象数目记为 $N(a_i'a_j)$. 我们还使用下面的记法，它们的含义是一目了然的：

$$N(a_ia_j\cdots a_k),$$
$$N(a'_ia'_j\cdots a'_k),$$
$$N(a'_ia_j\cdots a_k),$$

其他.

在例 7.1 中，A 中的对象是 18 个人，所以 $N=18$. 设性质 a_1 是有编程技术，而性质 a_2 是有统计技术. 那么

$$N(a_1)=10,\quad N(a_2)=5,\quad N(a_1a_2)=2.$$

⊖ 对维恩图不熟习的读者应该参考有限数学的教材，例如参考文献：Goldstein，Schneider 和 Siegel[2001]或 Mizrahi 和 Sullivan[1999].

根据计算，得

$$N(a'_1a'_2) = 5.$$

我们还可以计算 $N(a'_1a_2)$ 和 $N(a_1a'_2)$．前者是有统计技术但没有编程技术的人的数目，它等于 $5-2=3$.

404

在例 7.1 中的计算中，我们使用了下面的 $N(a'_1a'_2)$ 的公式：

$$N(a'_1a'_2) = N - N(a_1) - N(a_2) + N(a_1a_2). \tag{7.1}$$

注意，某些对象被包容太多次，所以其中的某些必须被排斥．等式(7.1)是著名的容斥原理(principle of inclusion and exclusion)的一种特殊情况．包容和排斥对象的过程分别对应于等式(7.1)中的加法和减法．我们为 $N(a'_1a'_2a'_3)$ 开发类似的原理，这是既没有性质 a_1 又没有性质 a_2 也没有性质 a_3 的对象数目．这一原理的解释见图 7.2 中的维恩图．首先，我们使 A 包容所有对象(数目总共为 N)．然后，我们排除那些有性质 a_1 的所有对象、有性质 a_2 的所有对象和有性质 a_3 的所有对象．因为某些对象可能有多个性质，所以我们有必要加回被过多排斥掉的对象．我们加回(包容)有两个性质的对象，这些对象对应于图 7.2 阴影线的区域．这时，我们又多加回了若干对象，即有三个性质的那些对象，这些对象对应于图 7.2 中交叉阴影线的区域．这些对象必须被排除．对这一推理的结果加以公式化，我们得到下面的公式：

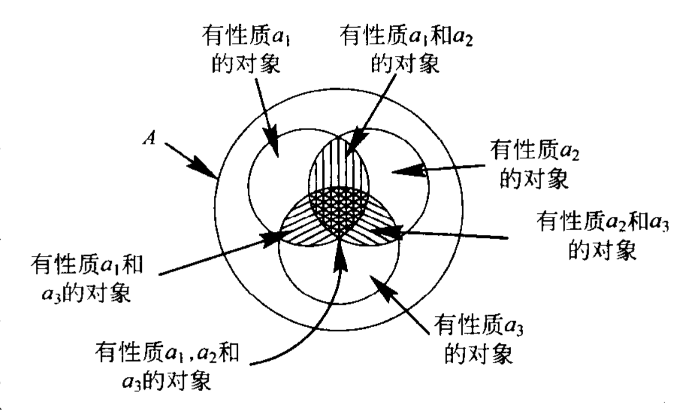


图 7.2　说明容斥原理的维恩图

$$\begin{aligned} N(a'_1a'_2a'_3) = & N - N(a_1) - N(a_2) - N(a_3) + N(a_1a_2) \\ & + N(a_1a_3) + N(a_2a_3) - N(a_1a_2a_3). \end{aligned} \tag{7.2}$$

一般地，没有任意 r 个性质的对象数量都可通过扩展公式(7.2)得到．这个一般公式称为*容斥原理*(principle of inclusion and exclusion)．我们给出的容斥原理的这一形式是 Sylvester 在 19 世纪中期发现的．这一原理的另一种形式是由 De Moivre[1718]较之 Sylvester 早几年发现的．这一原理由下面的定理给出．

定理 7.1(容斥原理)　*如果 N 是集合 A 中的对象的数量，集合 A 中没有性质 a_1，a_2，…，a_r 的对象数量由下式给出：*

$$\begin{aligned} N(a'_1a'_2\cdots a'_r) = & N - \sum_i N(a_i) + \sum_{i\neq j} N(a_ia_j) \\ & - \sum_{\substack{i,j,k \\ \text{互不相同}}} N(a_ia_ja_k) \pm \cdots + (-1)^r N(a_1a_2\cdots a_r). \end{aligned} \tag{7.3}$$

在(7.3)式中，第一个和是对所有取自 $\{1, 2, \cdots, r\}$ 的 i 求和．而第二个和是对所有无序对 $\{i, j\}$ 求和，其中 i 和 j 取自 $\{1, 2, \cdots, r\}$，$i\neq j$．第三个和是对于所有无序三元组 $\{i, j, k\}$ 求和，其中 i，j 和 k 取自 $\{1, 2, \cdots, r\}$，i，j 和 k 互不相同．一般项是 $(-1)^t$ 乘以形如 $N(a_{i_1}, a_{i_2}, \cdots, a_{i_t})$ 的项的和，其中这种形式的和是对于所有无序 t 元组 $\{i_1, i_2, \cdots, i_t\}$ 求和，而 i_1，i_2，…，i_t 取自 $\{1, 2, \cdots, r\}$，且 i_1，i_2，…，i_t 互不相同．在本章的余下部分，我们将给出容斥原理的应用和不同版本的容斥原理．我们将在 7.1.2 节中给出定理 7.1 的证明．

405

【例 7.2 排放量测试】　对 50 辆汽车做氧化氮(NO_x)、碳氢化合物(HC)和一氧化碳等污染物

排放量的测试. 其中，一辆汽车的排放量超过所有三种污染物的环境标准. 三辆汽车 NO 和 HC 超标，两辆车 NO_x 和 CO 超标，一辆汽车 HC 和 CO 超标，六辆汽车 NO_x 超标，四辆汽车 HC 超标，三辆汽车 CO 超标. 有多少辆汽车符合所有三种污染物的环境标准? 我们设 A 是汽车的集合，设性质 a_1 是 NO_x 超标，性质 a_2 和性质 a_3 分别是 HC 超标和 CO 超标. 我们要计算 $N(a_1'a_2'a_3')$. 已知下面的信息:

$$N=50,\quad N(a_1a_2a_3)=1,\quad N(a_1a_2)=3,\quad N(a_1a_3)=2,$$
$$N(a_2a_3)=1,\quad N(a_1)=6,\quad N(a_2)=4,\quad N(a_3)=3.$$

因此，根据容斥原理，有

$$N(a'_1a'_2a'_3)=50-6-4-3+3+2+1-1=42.$$

■

7.1.2　定理 7.1 的证明[⊖]

定理 7.1 的证明思想很简单. (7.3)式的左边计数 A 中没有所有性质的对象数量. 我们将简单地证明没有所有性质的每一个对象在(7.3)式的右边正好被计数一次，至少有一个性质的每一个对象(作为结果)正好被计数 0 次. 假设一个对象没有所讨论的任何性质. 那么在计算 N 时它被计数一次，但是在$\sum N(a_i)$，$\sum N(a_ia_j)$等中却不被计数. 因此，它在(7.3)式的右边正好被计数一次. 假设一个对象正好有性质 a_1，a_2，…，a_r 中的 p 个性质，其中 $p>0$. 现在，在计算 A 中对象数目 N 时，它被计数 $1=\binom{p}{0}$次. 在它所具有的性质 a_i 的每一个表达式 $N(a_i)$中它都被计数 1 次，所以在$\sum N(a_i)$中正好被计数 $p=\binom{p}{1}$次. 这个对象以多少项 $N(a_ia_j)$被计数呢? 其答案是这个对象拥有的性质 a_i 和性质 a_j 的配对数量. 这个数量是由从 p 个性质中取出两个性质的方法数量给出的，即由$\binom{p}{2}$给出. 因此，这个对象在$\sum N(a_ia_j)$中正好被计数$\binom{p}{2}$次. 类似地，在$\sum N(a_ia_ja_k)$中，这个对象正好被计数$\binom{p}{3}$次，以此类推. 总之，这个对象在(7.3)式的右侧被计数的次数由下式给出:

$$\binom{p}{0}-\binom{p}{1}+\binom{p}{2}-\binom{p}{3}\pm\cdots+(-1)^r\binom{p}{r}. \tag{7.4}$$

因为 $p\leqslant r$，且为了方便，如果 $p<k$ 有$\binom{p}{k}=0$，(7.4)式变成

$$\binom{p}{0}-\binom{p}{1}+\binom{p}{2}-\binom{p}{3}\pm\cdots+(-1)^p\binom{p}{p}. \tag{7.5}$$

因为 $p>0$，定理 2.9 表明表达式(7.5)是 0，所以这个对象对(7.3)式右边的总贡献计数是 0. 这就是整个证明.

7.1.3　素数、密码学和筛

引起数学家兴趣的关于数的一个最早的问题是确定所有*素数*(prime number)的问题，所谓的素数就是大于 1 且只有 1 和它本身是其正因子的正整数. 最近，Agrawal，Kayal 和Saxena[2002] 给出了一个确定给定整数 $n>1$ 是素数还是合数(composite)(非素数)的确定性多项式时间算法. 在此之前，可用于测试一个整数是素数还是合数的算法只有非多项式时间算法、假设已知某些

⊖　本小节可以跳过. 然而，还是建议读者阅读一下本小节的内容.

406

猜测下的多项式时间算法和概率算法. 例如，如果通过概率算法无法确定一个数是合数，那么这个数很有可能是素数，即*工业等级素数*(industrial-grade prime). 这一称呼是一个很重要的称呼，因为在很多实际应用中都需要素数，而且一直在使用工业等级素数. 对于其他更多关于素数的测试方法，参见 Adleman 和 Huang[1992]、Adleman，Pomerance 和 Rumely[1983]、Goldwasser 和 Kilian[1986]、Miller[1976]、Rabin[1980]、Solovay 和 Strassen[1977].

素数的一个重要用途是在*密码学*(cryptography)中，这一领域是致力于隐藏信息的数学学科. 今天，密码学出现于诸如保证电子邮件的安全性、保护医疗记录的隐私性、保护电子商务交易的完整性以及保护数字领域的版权等方面. 例如，众所周知，每一个大于1的正整数能够唯一分解成素数的积. 这一点是密码学中最常用的“公钥”算法的关键，*RSA 算法*(RSA Algorithm)是 Rivest，Shamir 和 Adleman[1978]给出的. 长期以来，因数分解整数一直被认为是一个困难的计算问题. 人们普遍认为在相对于数的位数或表示数的数字数目的多项式时间内，一个整数是不能被分解的. 所以，人们普遍相信诸如 RSA 这样的依赖于因数分解难度的密码方法的安全性. 以问题的难度(即非多项式时间)为基础的密码学方法的基础之一是，在多项式时间内可能计算的内容的定义与使用的计算模型无关. 量子计算，使用以量子力学为基础的设备，就是这样的一个模型. 量子计算机与通常的计算机的操作方式不同，它利用原子的量子状态，并提供远远超出当前的并行超级计算机的计算能力. 然而，当前只存在这类计算机的模型. Shor[1997]证明了在量子计算模型下，一个数可以在多项式时间内被因数分解的一个令人注目的结果. 对于成为另一个密码学方法基础的离散对数问题，Diffie-Hellmn 法(参见 Diffle 和 Hellman[1976])，他得到了类似的结果. 因此，对于非常有实际效益的问题，量子模型有比传统计算模型快得多的算法. 这些结果不仅表明量子计算可以给计算注入新的力量，它也使人们对基于因数分解或离散对数的难度的密码系统的安全性产生怀疑(更多关于密码系统的内容，参见 Anshel，Anshel 和 Goldfeld[1999]、Koblitz[1994]、Rhee[1993]、Schneier[1995]、Seberry 和 Pieprzyk[1989]、Stallings[1999]). 我们将在9.2.5节重新讨论密码系统，而在9.3.2节特别讨论 RSA 密码系统.

407

数学的一个古老问题是确定所有素数. 人们认为，下面确定1和 N 之间的所有素数的过程要归功于希腊的 Erastothenes. 首先写出1与 N 之间的所有数. 抹掉1. 然后抹掉能被2整除且大于2的那些数. 搜索大于2且还没有被抹掉的第一个数，这个数是3. 抹掉能被3整除且大于3的那些数. 然后搜索大于3还没有被抹掉的第一个数，这个数是5，然后抹掉能被5整除且大于5的那些数. 以此类推. 当这个过程结束时所留下的数就是素数. 下面给出 $N=25$ 时的这个过程的步骤：

1. 抹掉1并抹掉能被2整除且大于2的那些数：

   ~~1~~ 2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ 9 ~~10~~ 11 ~~12~~ 13 ~~14~~ 15 ~~16~~ 17 ~~18~~ 19 ~~20~~ 21 ~~22~~ 23 ~~24~~ 25

2. 抹掉能被3整除且大于3的那些还没有被抹掉的数：

   ~~1~~ 2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ ~~9~~ ~~10~~ 11 ~~12~~ 13 ~~14~~ ~~15~~ ~~16~~ 17 ~~18~~ 19 ~~20~~ ~~21~~ ~~22~~ 23 ~~24~~ 25

408

3. 抹掉能被5整除且大于5的那些还没有被抹掉的数：

   ~~1~~ 2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ ~~9~~ ~~10~~ 11 ~~12~~ 13 ~~14~~ ~~15~~ ~~16~~ 17 ~~18~~ 19 ~~20~~ ~~21~~ ~~22~~ 23 ~~24~~ ~~25~~

整除7，11等等，不再删除数，所以所留下的数就是1与25之间的素数. 这个过程通常在蜡版上进行(Vilenkin[1971])，数不是被抹掉，而是被穿孔. 结果有些像筛子，因此，这个过程就被命名为 *Erastothenes 的筛子*(Sieve of Erastothenes). 一个基本问题是：1和 N 之间有多少个素

数？答案与下面问题的答案有密切的关系：在 1 和 N 之间有多少个数(除了 1)不能被 2，3，5，…整除？在下面的例子中，我们来看如何回答这一类型的问题.

【例 7.3】 在下列情况下，1 和 1000 之间有多少个整数？

(a)不能被 2 整除的.

(b)不能被 2 或 5 整除的.

(c)不能被 2、5 或者 11 整除的.

为了回答这些问题，我们考虑 1 到 1000 之间的 1000 个整数的集合，并设 a_1 是能被 2 整除的性质，a_2 是能被 5 整除的性质，a_3 是能被 11 整除的性质. 我们希望有下面的信息：

$$(a)N(a'_1);\quad (b)N(a'_1a'_2);\quad (c)N(a'_1a'_2a'_3).$$

我们已知 $N=1000$. 也知道

$$N(a_1)=500,$$

因为每隔一个整数就可以被 2 整除. 因此，

$$N(a'_1)=N-N(a_1)=500,$$

这给出(a)的答案. 接下来

$$N(a_2)=\frac{1}{5}(1000)=200.$$

同样，每隔 10 个整数能被 2 和 5 整除，所以

$$N(a_1a_2)=\frac{1}{10}(1000)=100.$$

因此，根据容斥原理有

[409]

$$N(a'_1a'_2)=1000-500-200+100=400,$$

这给出(b)的答案. 最后，我们有

$$N(a_3)=\frac{1}{11}(1000)=90.9.$$

当然，因为 $N(a_3)$ 是整数，这表明

$$N(a_3)=90.$$

简而言之，有：

$$N(a_3)=\lfloor 90.9\rfloor=90.$$

同样，每隔 22 个整数能被 2 和 11 整除，所以

$$N(a_1a_3)=\left\lfloor\frac{1}{22}(1000)\right\rfloor=\lfloor 45.5\rfloor=45.$$

类似地，

$$N(a_2a_3)=\left\lfloor\frac{1}{55}(1000)\right\rfloor=\lfloor 18.2\rfloor=18.$$

最后，每隔 110 个整数能被 2、5 和 11 整除，所以

$$N(a_1a_2a_3)=\left\lfloor\frac{1}{110}(1000)\right\rfloor=\lfloor 9.1\rfloor=9.$$

于是

$$N(a'_1a'_2a'_3)=1000-(500+200+90)+(100+45+18)-9=364.$$ ■

【例 7.4 与给定整数互质的整数数量】 两个整数*互质*(relatively prime)，如果它们没有大于 1 的公约数. 两个非互质的整数必定有素公约数(为什么). 因此，$45=3^2\cdot 5$ 和 $56=2^3\cdot 7$ 是互质的. 1 到 1000 之间有多少个整数与 1000 是互质的？因为 $1000=2^3\cdot 5^3$，整除 1000 的素数只

有 2 和 5. 所以，我们要寻找在 1 到 1000 之间不能被 2 或 5 整除的整数数量. 在例 7.3 中，我们已找到这个数是 400. ■

可以扩展前面例子中所述的方法. 欧拉 ϕ 函数(Euler's ϕ function)$\phi(n)$定义为 1 到 n 之间与 n 互质的整数数量.

定理 7.2 如果 n 是有如下唯一素数因子分解的整数：

$$n = p_1^{e_1} p_2^{e_2} \cdots p_r^{e_r},$$

那么

$$\phi(n) = n\Big(1-\frac{1}{p_1}\Big)\Big(1-\frac{1}{p_2}\Big)\cdots\Big(1-\frac{1}{p_r}\Big).$$

410

证明 一个 1 与 n 之间的整数与 n 不是互质的，当且仅当它能被 p_1 或 p_2 或…或 p_r 整除. 设 $A=\{1, 2, \cdots, n\}$且 a_i 是被 p_i 整除的性质. 那么可以通过计数 A 中是 $p_{i_1}p_{i_2}\cdots p_{i_m}$ 倍数的元素数目得到 $N(a_{i_1}a_{i_2}\cdots a_{i_m})$，即

$$N(a_{i_1}a_{i_2}\cdots a_{i_m}) = \left\lfloor \frac{n}{p_{i_1}p_{i_2}\cdots p_{i_m}} \right\rfloor = \frac{n}{p_{i_1}p_{i_2}\cdots p_{i_m}}.$$

因为 n 能被 $p_{i_1}, p_{i_2}, \cdots, p_{i_m}$ 整除. 根据容斥原理，有

$$\phi(n) = N(a'_1a'_2\cdots a'_r) = n - n\sum_i \frac{1}{p_i} + n\sum_{i\neq j}\frac{1}{p_ip_j} \mp \cdots + (-1)^r n\frac{1}{p_1p_2\cdots p_r}.$$

这一结果得自于对下面关系的观察：

$$(1+x_1)(1+x_2)\cdots(1+x_r) = \sum_{I\subseteq\{1,2,\cdots,r\}}\Big(\prod_{i\in I}x_i\Big).$$ ■

对于更多关于欧拉 ϕ 函数的内容和数论中的相关结果，参见如 Hardy 和 Wright[1980]等的著作；也可参见练习 32.

【例 7.5 妇女和鸡蛋、斐波那契数以及筛子】 下面是一个中世纪的古老难题(Reingold, Nievergelt 和 Deo[1977]). 一位年迈的妇女在去市场卖鸡蛋的路上被一个马夫撞倒. 因为所有的鸡蛋都打碎了，所以这个马夫提出给她赔偿损失. 这位老妇人却不记得她原来有多少鸡蛋. 但她却记得当她一次取两个鸡蛋时，最后剩下一个鸡蛋，而且当她一次取三个和一次取四个鸡蛋时也总是剩下一个鸡蛋. 然而，当她一次取出 5 个鸡蛋时却没有剩余的鸡蛋. 如果合理地假设她所有的鸡蛋个数最多不超过 25 个，那么有方法确定这位老妇人到底有多少个鸡蛋? 一个自然的解决方法是模仿 Erastothenes 的筛子. Erastothenes 的筛子基于寻找 1 到 n 之间下面所有集合内所有数的问题：

$$\{2k+1:k\geqslant 1\},\{3k+1,3k+2:k\geqslant 1\},\{5k+1,5k+2,5k+3,5k+4:k\geqslant 1\},$$
$$\{7k+1,7k+2,7k+3,7k+4,7k+5,7k+6:k\geqslant 1\},\cdots,$$
$$\{pk+1,pk+2,\cdots,pk+p-1:k\geqslant 1\},$$

其中 p 是小于或等于 n 的最大素数. 这位老妇人的问题可以类似地阐述为寻找 1 到 n 之间下面所有集合中所有整数的问题：

$$\{2k+1:k\geqslant 1\},\{3k+1:k\geqslant 1\},\{4k+1:k\geqslant 1\},\{5k:k\geqslant 1\}.$$

一个筛子可以用于解决 $n=25$ 时的这一问题，方法是列出 1 到 25 之间的所有整数，并抹掉所有形如 $2k$ 的整数、所有形如 $3k$ 或 $3k+2$ 的整数，所有形如 $4k$ 或 $4k+2$ 或 $4k+3$(前两种类型的整数可能已被抹掉)的整数，最后，抹掉所有形如 $5k+1$，$5k+2$，$5k+3$，$5k+4$ 的整数. 这位老妇人有多少鸡蛋呢? Reingold，Nievergelt 和 Deo[1977]给出了类似的筛子方法如何用于解决更复杂的问题，例如快速检测前 1 百万个斐波那契数，看哪些是完全平方数(只有两个，它们是

411

1 和 144). ■

7.1.4　概率情况

假设随机选取 1 到 1000 之间的一个整数. 它不能被 2、5 或 11 整除的概率是多少呢? 答案很简单. 考虑一个实验, 在这个实验中结果是 1 到 1000 之间的一个整数, 而且结果的几率均等. 根据前节的计算, 显示不能被 2, 5 或 11 整除的事件的结果数目是 364. 所以, 所讨论的概率是 364/1000=0.364.

更一般地, 假设我们考虑一个在样本空间的有限集合 S 内产生一个结果的实验. 设要考虑的事件是 E_1, E_2, …, E_r. 其中任何一个事件都不发生的概率是多少? 为了回答这个问题, 正如我们在本书所做的那样, 我们假设样本空间 S 内的所有结果的几率是均等的(然而, 可以证明得到本小节的主要结果无需做这样的假设). 设集合 A 是集合 S, 并设 a_i 是结果显示事件 E_i 的性质. 设 $p_{ijk\cdots}$ 是事件 E_i、E_j、E_k…发生的概率. 根据定理 7.1, 我们得出 E_1, E_2, …, E_r 中任何事件都不发生的概率 p 由下式给出:

$$p=\frac{N(a_1'a_2'\cdots a_r')}{n(S)}$$
$$=1-\frac{\sum N(a_i)}{n(S)}+\frac{\sum N(a_ia_j)}{n(S)}-\frac{\sum N(a_ia_ja_k)}{n(S)}\pm\cdots+(-1)^r\frac{N(a_1a_2\cdots a_r)}{n(S)},$$

其中, $n(S)$是 S 中的结果数量. 因此

$$p=1-\sum p_i+\sum p_{ij}-\sum p_{ijk}\pm\cdots+(-1)^r p_{12\cdots r}. \tag{7.6}$$

【例 7.6 被水覆盖的两个对映点】[⊖] 在大多数拓扑学书籍中都有这样一个引起人们好奇心的问题, 这个问题是要求读者证明在环绕地球的任意一个大圆上, 都存在有相同温度的两个对映点. 人们也可以给组合学家提出类似的问题. 众所周知, 地球的表面有超过一半的面积都被水覆盖. 证明在地球上有两个都被水覆盖的对映点. 设 X 表示地球上的任意一个点. 具体地, 我们只考虑整数纬度和经度点, 所以所有这样的点的集合是有限的. 我们还假设 $-X$ 表示与 X 对映的点. 考虑下面的事件:

412

$$E_1 = 点\ X\ 被水覆盖,$$
$$E_2 = 点\ -X\ 被水覆盖.$$

根据(7.6)式, 事件 E_1 和 E_2 都不发生的概率 p 等于 $1-(p_1+p_2)+p_{12}$, 其中 p_i 是事件 E_i 发生的概率, p_{12}是 E_1 和 E_2 都发生的概率. 因为 p_1, p_2 都大于 1/2, 对于 0 到 1 之间的数 p, p_{12}必定是正的. 因此, 一定存在具有两个性质的点 X, 即使得 X 和 $-X$ 都被水覆盖的 X. ■

7.1.5　可区分球和盒子的分装问题

在 2.10 节, 我们考虑了把 n 个可区分球放入 c 个可区分盒子的分装问题. 现在我们要问: 所有盒都非空的概率是多少? 设 S 是把球放入盒子的分配的集合, 且设 E_i 是第 i 个盒子为空的事件. 如上定义 A 和 a_i. 这时, $N(S)=c^n$, $N(a_i)=(c-1)^n$, $N(a_ia_j)=(c-2)^n$, $N(a_ia_ja_k)=(c-3)^n$, …, 另外, 存在$\binom{c}{1}$种方法选择性质 a_i, $\binom{c}{2}$种方法选择性质 a_i 和 a_j, 以此类推. 因此, n 个可区分球放入 c 个可区分盒子且没有盒子为空的分配方法的数量由下式给出:

$$c^n-\binom{c}{1}(c-1)^n+\binom{c}{2}(c-2)^n-\binom{c}{3}(c-3)^n\pm\cdots+(-1)^c\binom{c}{c}(c-c)^n,$$

⊖ 本例子取自 Shen[1998].

上述表达式等于

$$\sum_{t=0}^{c}(-1)^t\binom{c}{t}(c-t)^n. \tag{7.7}$$

那么，所有盒子都非空的概率由下式给出：

$$\frac{c^n-\binom{c}{1}(c-1)^n+\binom{c}{2}(c-2)^n-\binom{c}{3}(c-3)^n+\cdots+(-1)^c\binom{c}{c}(c-c)^n}{c^n},$$

上述表达式等于

$$\sum_{t=0}^{c}(-1)^t\binom{c}{t}\left(1-\frac{t}{c}\right)^n, \tag{7.8}$$

因为

$$\frac{(c-t)^n}{c^n}=\left(1-\frac{t}{c}\right)^n.$$

【例 7.7 快餐奖品】 假设一家快餐店在儿童套餐中放入三种不同的玩具，一份套餐放入一个玩具. 如果我们买六份儿童套餐，而且每一种玩具在任意一份套餐里的几率是均等的，得到所有三种玩具的概率是多少？我们想象把 $n=6$ 个球，或玩具放入 $c=3$ 个盒子或玩具类型里. 把玩具放入到玩具类型中且使得没有盒子(或类型)为空的方法数量由等式(7.7)给出如下：

$$3^6-\binom{3}{1}\cdot 2^6+\binom{3}{2}\cdot 1^6-\binom{3}{3}\cdot 0^6=540.$$

413

这一事件发生的概率是 $540/3^6=0.741$. 这一概率也可以通过等式(7.8)计算如下：

$$1-\binom{3}{1}\cdot\left(\frac{2}{3}\right)^6+\binom{3}{2}\cdot\left(\frac{1}{3}\right)^6-\binom{3}{3}\cdot 0^6=0.741.$$

■

7.1.6 色多项式

在 3.4 节，我们介绍了图的色多项式的概念. 容斥原理可以用于计算色多项式. 值得注意的是同一种计数问题是如何用多种方法解决的. 在本章的多个地方，我们把容斥原理运用于计数前面我们用其他方法计数的数量. 例如，考虑图 7.3 中的图. 我们考虑用 x 种颜色或少于 x 种颜色着色 G 的顶点的所有可能着色. 我们甚至允许由一条边连接的两个顶点得到相同颜色的着色，但是我们称这样的着色是不适宜的(improper)，而所有其他着色为适宜的(proper). 考虑用 x 种颜色或少于 x 种颜色的图 G 的所有着色的集合，不适宜着色和适宜着色的集合. 存在 x^4 种这样的着色，因为 4 个顶点中的每一个都可能够被 x 种颜色中的任意一种颜色着色. 我们将为图 G 的每一条边引入一个性质 a_i. 因此，

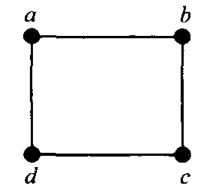


图 7.3 一个图

a_1 是顶点 a 和 b 得到相同颜色的性质，

a_2 是顶点 b 和 c 得到相同颜色的性质，

a_3 是顶点 c 和 d 得到相同颜色的性质，

a_4 是顶点 d 和 a 得到相同颜色的性质.

为了计算使用 x 种颜色或少于 x 种颜色的 G 的(适宜)着色数量 $P(G, x)$，我们必须计算 $N(a_1'a_2'a_3'a_4')$. 我们有 $N(a_1)=x^3$，因为，存在 x 种着色 a 和 b 的选择，存在 x 种着色 c 的选择，存在 x 种着色 d 的选择(回想一下，不适宜着色是允许的). 类似地，有

$$N(a_2)=N(a_3)=N(a_4)=x^3.$$

其次，

$$N(a_1a_2)=x^2.$$

因为 a 和 b 必须得到相同的颜色，而且 b 和 c 也必须得到相同的颜色. 因此，对于 a，b 和 c 得到一种颜色存在 x 种选择，那么 d 得到的颜色存在 x 种选择. 类似的推理证明

$$N(a_1a_3)=N(a_1a_4)=N(a_2a_3)=N(a_2a_4)=N(a_3a_4)=x^2.$$

类似地，有

$$N(a_1a_2a_3)=N(a_1a_2a_4)=N(a_1a_3a_4)=N(a_2a_3a_4)=x,$$

因为在所有这些情况下，所有顶点必须得到相同的颜色. 这一推理也使我们得到下面的结论：

414

$$N(a_1a_2a_3a_4)=x.$$

因此，根据容斥原理，有

$$\begin{aligned}P(G,x)&=N(a'_1a'_2a'_3a'_4)=x^4-4x^3+6x^2-4x+x\\&=x^4-4x^3+6x^2-3x.\end{aligned}$$

可以使用第 3 章的方法检验这一计算.

我们如下扩展这个例子. 假设 G 是任意图，而且我们希望计算 $P(G,x)$. 考虑用 x 种颜色或少于 x 种颜色的 G 的顶点的所有着色(适宜着色和不适宜着色)的集合 A. 对于每一条边 i，设 a_i 是边 i 的端部顶点得到相同颜色的性质. 假设 $|V(G)|=n$，且 $|E(G)|=r$. 那么 $N=|A|=X^n$，且

$$P(G,x)=N(a'_1a'_2\cdots a'_r).$$

因此

$$P(G,x)=x^n-\sum N(a_i)+\sum N(a_ia_j)\mp\cdots+(-1)^e\sum N(a_{i_1}a_{i_2}\cdots a_{i_e})+\cdots. \tag{7.9}$$

考虑项 $N(a_{i_1}a_{i_2}\cdots a_{i_e})$. 假设 H 是由 G 的所有顶点组成且有边 i_1，i_2，…，i_e 的 G 的子图. 包含 G 的所有顶点的子图 H 在第 3 章中称为**支撑子图**(spanning subgraph). 注意，满足性质 a_{i_1}，a_{i_2}，…，a_{i_e} 的 G 的(适宜的或不适宜的)着色等价于满足性质 a_{i_1}，a_{i_2}，…，a_{i_e} 的 H 的(适宜的或不适宜的)着色. 现在，在 H 的这样的一个着色中，H 的任意一个连通分支的所有顶点必须有相同的颜色. 对 H 的一个分支的着色的选择是随机的. 因此，使用 x 种颜色或少于 x 种颜色且满足性质 a_{i_1}，a_{i_2}，…，a_{i_e} 的 G 的顶点的着色数量由 $x^{c(H)}$ 给出，其中 $c(H)$ 是 H 的连通分支的数量.

每一个有 e 条边和 c 个连通分支的支撑子图 H 对应于性质 a_{i_1}，a_{i_2}，…，a_{i_e} 的某个集合，而且为(7.9)式的右边提供一个项 $(-1)^e x^c$. 因此，我们有下面的定理.

415

定理 7.3[⊖] *如果 G 是一个图且 $h(e,c)$ 是有 e 条边和 c 个分支的支撑子图的数量，那么*

$$P(G,x)=\sum_{e,c}(-1)^e h(e,c)x^c.$$

在这一定理中，注意，如果 $c=n$ 则 $h(0,c)=1$，否则 $h(0,c)=0$. 注意这一定理给出 $P(G,x)$ 是多项式的快速证明. 在图 7.3 的例子中，有下面的结果，这一结果在图 7.4 中说明：

$$h(4,1)=1,\quad h(3,1)=4,\quad h(2,2)=6,\quad h(1,3)=4,\quad h(0,4)=1,$$

而且，对于其他的 e 和 c，$h(e,c)=0$. 因此，我们有

⊖ 这一定理是由 Birkhoff[1912](在研究地图中出现的图时)发现的，且首次由 Whitney[1932]根据容斥原理得到证明.

$$P(G,x) = (-1)^4 h(4,1)x + (-1)^3 h(3,1)x + (-1)^2 h(2,2)x^2 + (-1)^1 h(1,3)x^3 + (-1)^0 h(0,4)x^4$$
$$= x - 4x + 6x^2 - 4x^3 + x^4 = x^4 - 4x^3 + 6x^2 - 3x,$$

这一多项式与我们上面的计算一致.

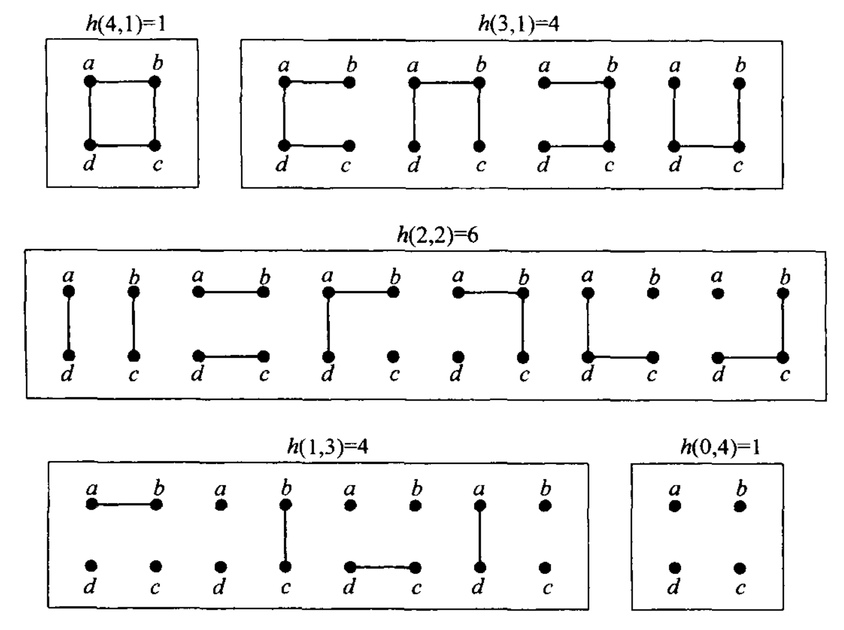


图 7.4 图 7.3 中的图的支撑子图

416

7.1.7 错位排列

我们在 6.1.3 节介绍了错位排列是一个排列，其中任意对象都不在其正确的位置上. 我们给出如何使用容斥原理计算 n 个对象集合的错位排列的数量 D_n. 考虑 n 个对象的所有排列的集合 A. 设 a_i 是对象 i 被放置在第 i 位置上的性质. 因此

$$D_n = N(a'_1 a'_2 \cdots a'_n).$$

我们有

$$N = n!$$

个排列. 还有

$$N(a_i) = (n-1)!,$$

因为对象 i 返回其原来位置的排列数等价于其余对象的排列数. 类似地，有

$$N(a_i a_j) = (n-2)!,$$

且有

$$N(a_{i_1} a_{i_2} \cdots a_{i_t}) = (n-t)!.$$

因此，有

$$\sum N(a_{i_1} a_{i_2} \cdots a_{i_t}) = \binom{n}{t}(n-t)!,$$

因为对于性质 $a_{i_1} a_{i_2} \cdots a_{i_t}$ 有 $\binom{n}{t}$ 种选择. 根据容斥原理有

$$D_n = N - \sum N(a_i) + \sum N(a_i a_j) - \sum N(a_i a_j a_k) \pm \cdots + (-1)^n N(a_1 a_2 \cdots a_n)$$
$$= n! - \binom{n}{1}(n-1)! + \binom{n}{2}(n-2)! - \binom{n}{3}(n-3)! \pm \cdots + (-1)^n \binom{n}{n}(n-n)!.$$

化简后，我们得

$$D_n = n! - \frac{n!}{1!(n-1)!}(n-1)! + \frac{n!}{2!(n-2)!}(n-2)! - \frac{n!}{3!(n-3)!}(n-3)! \pm \cdots + (-1)^n \frac{n!}{n!(n-n)!}(n-n)!$$

$$= n!\left[1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} \pm \cdots + (-1)^n \frac{1}{n!}\right],$$

417 这一结果与我们前面所看到的一致.

7.1.8　计数组合

在 5.3 节，我们通过生成函数的方法研究了各种计数问题. 这里，我们注意到，容斥原理也可以用于这类问题. 我们通过一个例子来说明这一方法. 假设我们正在做一项调查，而且我们考虑要对三名老师、四名水管工人和六名汽车工人做访谈. 假设我们考虑做相同工作的两名人员是不可区分的，而且我们寻找选择 11 个人进行访谈的方法数量. 根据第 5 章的方法，这一方法数量可以通过确定下面生成函数中 x^{11} 的系数来计算：

$$(1+x+x^2+x^3)(1+x+x^2+x^3+x^4)(1+x+x^2+x^3+x^4+x^5+x^6). \qquad (7.10)$$

然而，我们下面使用与此不同的方法计算.

考虑有无限多名老师、水管工人和汽车工人的情况，而且考虑选择 11 个人进行访谈的所有方法组成的集合 A. 对于集合 A 的特定元素，我们说它满足性质 a_1，如果它至少使用 4 名老师；满足性质 a_2，如果它至少使用 5 名水管工人；满足性质 a_3，如果它至少使用 7 名汽车工人. 我们要计数集合 A 中所有不满足任何性质 a_i 的元素的数量；因此我们要寻找 $N(a_1'a_2'a_3')$. 为了计算这个数量，注意根据定理 5.4，有

$$N = |A| = \binom{3+11-1}{11} = \binom{13}{11} = 78.$$

$N(a_1)$等于多少呢？注意，一个选择满足 a_1 当且仅当它至少有 4 名老师. 当每种人员有无限多个时，这样的选择相当于任选 7 个人，所以，根据定理 5.4，其方法数量为

$$\binom{3+7-1}{7} = \binom{9}{7} = 36.$$

因此，

$$N(a_1) = 36.$$

类似地，当每种人员有无限多个时，满足 a_2 的选择相当于任选 6 个人，所以

$$N(a_2) = \binom{3+6-1}{6} = \binom{8}{6} = 28.$$

最后，有

$$N(a_3) = \binom{3+4-1}{4} = \binom{6}{4} = 15.$$

接下来，同时满足 a_1 和 a_2 的选择至少有 4 名老师且至少有 5 名水管工人，当每种人员的数量不受限制时，这一选择相当于选出 2 个人. 因此

418

$$N(a_1a_2) = \binom{3+2-1}{2} = \binom{4}{2} = 6.$$

类似地，当每一类型的人员的数量不受限制时，同时满足 a_1 和 a_3 的选择相当于不选择任何人；因为不选择任何人正好有一种方法. 所以有

$$N(a_1a_3)=1.$$

同样，对于至少有5名水管工人和至少有7名汽车工人的情况，没有选择11个人的方法，所以有

$$N(a_2a_3)=0.$$

类似地，有

$$N(a_1a_2a_3)=0.$$

因此，根据容斥原理，所求的选择数量等于

$$78-(36+28+15)+(6+1+0)-0=6.$$

通过生成函数(7.10)式中x^{11}的系数很容易检验这一结果.

7.1.9 车多项式⊖

在例5.10、例5.14和例6.15中，我们研究了由可接受(深色)格子和不可接受格子组成的棋盘B的车多项式的概念. 如果如图5.1和图5.2所示的那样，棋盘B中深色格子占多数的话，那么考虑B的补棋盘(complementary board)B'很有用，棋盘B'是通过交换B的可接受格子和不可接受格子而得的. 假设B是一个$n\times m$的棋盘，我们感兴趣的是$r_n(B)$，这个数字是把n个不能吃掉的车放入B的可接受格子上的方法数量. 我们将证明可以通过$R(x, B')$得到$r_n(B)$，$R(x, B')$是补棋盘的车多项式，而不是从棋盘B的车多项式$R(x, B)$得到$r_n(B)$. 前者的车多项式是基于有较少深色格子的棋盘的，所以更容易计算. 我们可以假设$n\leqslant m$. 因为如果$n>m$，那么$r_n(B)=0$. 注意，对于$j<n$，我们不去计算$r_j(B)$，只计算$j=n$的特殊情况. 7.2节练习37要求读者把这一结果扩展到$j\leqslant n$时的任意$r_j(B)$.

n个不吃车对$n\times m$棋盘的一个分配指的是每一个车被放入一个格子，格子可以是可接受的也可以是不可接受的，且任意两个车不在同一行和同一列上. 那么，存在：

$$P(m,n)=m(m-1)\cdots(m-n+1)$$

种可能的分配. 因为，我们选择第一行的m个位置中的一个位置，然后在第二行的$m-1$个位置中选择一个位置，…，最后，在第n行的$m-n+1$个位置中选择一个位置. 设A是棋盘B的所有可能分配的集合，并设a_i是一个分配在第i列的一个不可接受格子中有一个车的性质. 那么$r_n(B)$等于没有任何性质a_1，a_2，…，a_m的分配数量. 我们利用容斥原理计算这个数. 419

给定t，我们来看如何计算

$$\sum N(a_{i_1}a_{i_2}\cdots a_{i_t}),$$

当然，其中t必定至多等于m. 注意，如果$t>n$，则上述和等于0，因为在t个不同的列上不可能存在车的分配，更不可能在t个不同列上的不可接受格子中存在车的分配. 如果$t\leqslant n$，那么考虑补棋盘B'. B的n个不吃车的分配对应于棋盘B'可接受格子的t个不吃车的分配，其中B的t个列的每一列中都有一个车在被禁止位置上，这样的分配共有$r_t(B')$种方法可以实现，于是其余的$n-t$个车可以随意放置在剩余的$m-t$列的任何位置上，这有$P(m-t, n-t)$种放置方法. 因此，对于$t\leqslant n$，有

$$\sum N(a_{i_1}a_{i_2}\cdots a_{i_t})=P(m-t, n-t)r_t(B').$$

根据容斥原理，我们得到结论：

$$\begin{aligned}r_n(B)=&P(m,n)-P(m-1,n-1)r_1(B')\pm\cdots\\&+(-1)^tP(m-t,n-t)r_t(B')\pm\cdots\\&+(-1)^nP(m-n,0)r_n(B').\end{aligned}\tag{7.11}$$

⊖ 本小节可以跳过.

把这一结果运用于图 5.1 中的 5×5 棋盘 B，计算 $r_5(B)$. 补棋盘 B' 如图 7.5 所示. 根据 5.1 节练习 17 的化简结果和例 5.14，可证

$$R(x,B') = 1+6x+11x^2+6x^3+x^4. \qquad (7.12)$$

使用(7.12)式时，注意到 $P(a,\ a)=a!$，同时运用(7.11)式，我们有

$$r_5(B) = 5!-4!(6)+3!(11)-2!(6)+1!(1)-0!(0) = 31.$$

B'

图 7.5 图 5.1 中的棋盘 B 的补棋盘 B'

本节练习

1. 在一座城市有三个收费有限电视频道 A，B 和 C. 下面的结果是对该城市住户的一次调查而得到的：20%的住户使用 A，16%的住户使用 B，14%的住户使用 C，8%的住户使用 A 和 B，5%的住户使用 A 和 C，4%的住户使用 B 和 C，2%的住户使用 A，B 和 C. 不使用任何频道的住户所占的百分比是多少？ [420]
2. 在一次实验中，有控制和非控制两种处理方式. 有 3 个控制器和 80 个实验单元或模块. 每个控制器用于 25 个模块，每一对控制器用于相同模块 11 次，而且所有 3 个控制器同时用于相同模块 12 次. 有多少个模块没有使用控制器？（我们将在第 9 章详细地研究实验设计的类似条件.）
3. 一家烟草公司调查了 200 000 个人. 根据这家公司的报告，其中有 130 000 个是男性. 被调查者中有 90 000 人是烟民，其中有 10 000 人已患有癌症. 然而，接受调查的 200 000 人中，有 7000 名男性癌症患者，8000 名吸烟癌症患者，5000 名男性烟民. 最后，有 1000 男性烟民患有癌症. 这些调查中有多少女性非烟民且非癌症患者呢？这一烟草公司的调查有错误吗？
4. 有 800 人接受下列疾病的免疫性检查：肺结核、风疹和天花. 这 800 人中，发现有 350 人对肺结核有免疫力，450 对风疹有免疫力，350 人对天花有免疫力，150 人对肺结核和风疹有免疫力，200 人对风疹和天花有免疫力，250 人对肺结核和天花有免疫力，100 人对肺结核和风疹有免疫力，但对天花没有免疫力. 有多少人对这三种疾病都没有免疫力呢？
5. (a)假设在被调查的 1000 家住户中，有 30 家既没有锻炼用自行车也没有踏步车，有 50 家只有锻炼用自行车，有 60 家只有踏步车. 有多少家两者都有？

   (b)数字 0，1，2…9 的排列中有多少种排列的第一位数字大于 2，最后一位数字小于 7？

   (c)长度为 10 的 DNA 序列中各基 A，G，C 和 T 都至少有一个的序列有多少个？
6. 确定集合 A 中至少有性质 a_1，a_2，…，a_r 之一的对象个数的表达式.
7. 120 份水样接受是否含有下面三种化学物质的检查：汞、砷和铅. 这 120 份水样中，发现 17 份含有汞，15 份含有砷，14 份含有铅，10 份含有汞和砷，7 份含有砷和铅，15 份含有汞和铅，5 份含有汞和砷，但不含铅. 有多少水样至少含有一种化学物质？
8. 在检查站接受检查的 100 辆汽车中，9 辆有头灯缺损；8 辆有刹车缺损；7 辆有喇叭缺损；2 辆有风挡刮水器缺损；4 辆有头灯和刹车缺损；3 辆有头灯和喇叭缺损；2 辆有头灯和风挡刮水器缺损；3 辆有刹车和喇叭缺损；没有汽车有刹车和风挡刮水器缺损；1 辆有喇叭和风挡刮水器缺损；1 辆有头灯、刹车和喇叭缺损；1 辆有头灯，喇叭和风挡刮水器缺损；没有汽车有任意其他缺损组合. 确定至少有一种所讨论的缺损的汽车的数量.
9. 一个学校总共有 100 名学生参加考试. 其中，38 名学生参加法语考试，45 名学生参加物理考试，28 名学生参加数学考试，25 名学生参加历史考试，22 名学生参加法语和物理考试，23 名学生参加法语和数学考试，10 名学生参加物理和数学考试，1 名学生参加法语和历史考试，21 名学生参加物理和历史考试，14 名学生参加数学和历史考试，11 名学生参加法语、物理和数学考试，8 名学生参加法语、物理和历史考试，6 名学生参加法语、数学和历史考试，6 名学生参加物理、数学和历史考试，5 名学生参加所有四门课程的考试. 有多少学生至少参加所讨论的科目中一门课程的考试？ [421]
10. 一名故障检修员已查明一台计算机上有 3 个文件是潜在的问题源头. 在一次测试中，他使用每一个文件 12 次，同样测试中每一对文件一起使用 6 次，同样测试中 3 个文件一起使用 4 次. 在 8 次测试中，

没有使用任何文件．他一共进行了多少次测试？

11. 1～10 000 之间(包括 1 和 10 000)包含多少个不能被 5、7 和 11 整除的整数？

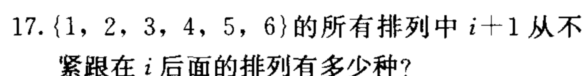
12. 1～600 之间(包括 1 和 600)包含多少个不能被 2、3 和 5 整除的整数？

13. 1～600 之间(包括 1 和 600)包含多少个不能被 2、3、5 和 7 整除的整数？

14. 一周内有 9 起事故发生．写出计算每天至少有一起事故发生的概率的表达式．

15. 一本书的 5 页上总共有 6 个印刷错误．每页至少有一个印刷错误的概率是多少？

16. 20 个光粒子撞击到有 9 个细胞的视网膜上．至少有一个细胞没有被光粒子撞击的概率是多少？



17. {1，2，3，4，5，6}的所有排列中 $i+1$ 从不紧跟在 i 后面的排列有多少种？

18. 使用容斥原理(不是定理 7.3)确定图 7.6 中的每一个图的色多项式．

19. 使用定理 7.3 确定图 7.6 中的每一个图的色多项式．

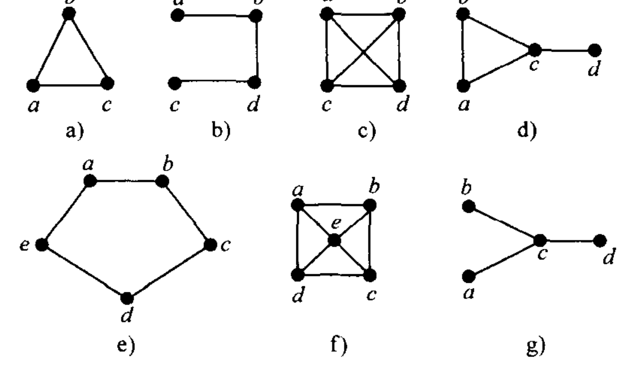


图 7.6　7.1 节练习的图

20. 星型图 $S(1, n)$是由 1 个中心顶点和 n 个邻居顶点组成的且没有其他边的图．图 7.7 给出若干个星型图．使用 7.1.6 节的方法确定 $S(1, n)$的色多项式． 422

21. 使用容斥原理计数进行下列选择的方法数量：

(a)从 4 个 a、4 个 b 和 5 个 c 的集合中选出 8 个元素的方法数量．

(b)从 3 个 a、4 个 b 和 5 个 c 的集合中选出 9 个元素的方法数量．

(c)从 6 个 a、6 个 b 和 4 个 c 的集合中选出 12 个元素的方法数量．

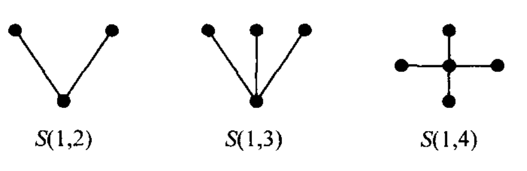


图 7.7　若干星型图

22. 验证等式(7.12)．

23. 假设 B 是所有方格都可接受的 $n\times n$ 棋盘．使用等式(7.11)计算 $r_n(B)$．

24. 使用等式(7.11)和更早的化简结果计算图 5.2 中的棋盘 B 的 $r_5(B)$．

25. 假设 $r=3$ 且一个对象正好有两个性质．在下列计算中该对象被计数了多少次？

(a)$\sum N(a_i)$．　　(b)$\sum N(a_ia_j)$．

26. 假设 $r=8$ 且一个对象正好有三个性质．在下列计算中该对象被计数了多少次？

(a)$\sum N(a_i)$．　(b)$\sum N(a_ia_j)$．　(c)$\sum N(a_ia_ja_k)$．　(d)$\sum N(a_ia_ja_ka_l)$．

27. 假设 d 个可区分 CD 被放置到 n 个可区分 CD 播放器中．若干 CD 可以放在一个播放器中．我们认为把 CD_5 放在播放器 1 中与把 CD_5 放在播放器 2 中是不同的，而且把 CD_5 放在播放器 1 中与把 CD_6 放在播放器 1 中也是可区分的等等．假设分配各 CD 使得各播放器都不是空的．有多少种方法可以做到这一点？

28. 使用包含和排斥确定下面方程的解的数量：

$$x_1+x_2+x_3=15,$$

其中，每一个 x_i 是非负整数且 $x_i\leqslant 7$．

29. 使用包含和排斥确定下面方程的解的数量：

$$x_1+x_2+x_3+x_4=18,$$

其中，每一个 x_i 是正整数且 $x_i\leqslant 8$．

30. 确定由数字表{0，1，2，…，9}上的 n 个数字组成的且数字 1，2 和 3 都至少出现一次的代码字的数量．

[423] 31. (a)确定从5元素集合到3元素集合的满射函数[⊖]的数量.

(b)假设 m 和 n 是正整数. 确定从 m 元素集合到 n 元素集合的满射函数的数量的公式.

32. 回想一下，每一个正整数 n 都可以唯一地写成素数的幂的积的形式，即

$$n = p_1^{e_1} p_2^{e_2} \cdots p_r^{e_r},$$

其中 p_1，p_2，…，p_r 是互不相同的素数，且对于所有的 i，有 $e_i \geqslant 1$. 如下定义默比乌斯函数 $\mu(n)$：

$$\mu(n) = \begin{cases} 1 & \text{若 } n = 1 \\ 0 & \text{若对某个 } i\text{，有 } e_i > 1 \\ (-1)^r & \text{若所有的 } e_1, e_2, \cdots, e_r \text{ 都等于 } 1. \end{cases}$$

因此，$\mu(100)=0$，因为 2^2 是100的因子，同样，$\mu(30)=\mu(2\cdot 3\cdot 5)=(-1)^3=-1$.

(a)根据容斥原理，证明

$$\sum_{d|n}\mu(d) = \begin{cases} 1 & \text{若 } n = 1 \\ 0 & \text{若 } n > 1, \end{cases} \tag{7.13}$$

其中，(7.13)式的和是对能够整除 n 的所有整数 d 求和. 例如，

$$\begin{aligned} \sum_{d|12}\mu(d) &= \mu(1)+\mu(2)+\mu(3)+\mu(4)+\mu(6)+\mu(12) \\ &= 1+(-1)+(-1)+0+(-1)^2+0 \\ &= 0. \end{aligned}$$

(b)假设 f 和 g 是函数，使得

$$f(n) = \sum_{d|n} g(d).$$

根据部分(a)的结果，证明

$$g(n) = \sum_{d|n} f\left(\frac{n}{d}\right)\mu(d). \tag{7.14}$$

等式(7.14)称为*默比乌斯反演公式*(Moebius inversion formula). 对于在组合数学中非常重要的这一公式的一般形式，参见 Rota[1964](也可参见 Berge[1971]、Hall[1986]和 Liu[1972]).

(c)证明：如果 $\phi(n)$ 是欧拉 ϕ 函数，那么

$$n = \sum_{d|n}\phi(d).$$

(d)得出结论：

$$\phi(n) = \sum_{d|n}\mu(d)\,\frac{n}{d}.$$

(e)证明：

[424]

$$\phi(p^e) = p^e\left(1-\frac{1}{p}\right).$$

33. (Cohen[1978]) n 名绅士中的每一名都存放帽子和雨伞. 帽子被随机还回，然后雨伞也独立地随机还回. 没有人拿到自己的帽子和雨伞的概率是多少?

34. 练习34和练习35考虑的是对某些模式进行限制的排列. 我们说，*模式 uv 不出现在*$\{1, 2, \cdots, n\}$的排列 $j_1j_2j_3\cdots j_n$ 中，意味着 j_ij_{i+1} 不会是 uv. 类似地，说*模式 uvw 不出现*意味着 $j_ij_{i+1}j_{i+2}$ 不会是 uvw. 设 b_n 是集合$\{1, 2, \cdots, n\}$的排列的数量，其中模式12，23，…，$(n-1)n$ 不出现. 确定 b_n.

35. 确定集合$\{1, 2, 3, 4, 5, 6\}$的排列的数量，其中模式125和34都不出现(参见练习34).

36. 排列字母 a，a，b，b，c，c，c，d，d，确定使得同一字母不连续出现的方法数量.

37. 数字表$\{0, 1, 2\}$上的长度为9、每一个数字都使用三次且任意三个连续数字不同的代码字有多少个?

38. 有两个A、两个U、两个C和两个G且没有重复基的RNA链有多少种?

39. 在研究整数划分时(5.3节练习12～练习16)，我们令 $p^*(k)$ 为有不同整数的 k 的划分的数量，$p_0(k)$ 为

⊖ 集合 A 到集合 B 的函数 f 是满射，当且仅当对于任意的 $b\in B$，存在 $a\in A$，使得 $f(a)=b$. ——译者注

有奇数个整数的 k 的划分的数量.

(a)定义一个集合 A 和 A 的元素的性质 a_i 和 b_i，使得

$$p_0(k) = N(a'_1a'_2\cdots),$$

且

$$p^*(k) = N(b'_1b'_2\cdots).$$

(b)证明可以选择(a)中的 A，a_i 和 b_i，使得对于所有的 k，

$$N(a_{i_1}a_{i_2}\cdots a_{i_k}) = N(b_{i_1}b_{i_2}\cdots b_{i_k})$$

由此得到结论 $p_0(k)=p^*(k)$(5.3 节练习 15 中已经通过生成函数推导出了这一结果).

40. (Shen[1998])一个球被涂上两种颜色：其表面的 10%是白色，其余的表面涂成黑色. 证明这一球有一个内接立方体，使得其所有 8 个顶点都是黑色[提示：考虑例 7.6].

7.2 正好有 m 个性质的对象数量

7.2.1 主要结果及其应用

我们返回到有 N 个对象的集合的一般情况，这个集合中的每一个对象可能有也可能没有 r 个不同性质 a_1，a_2，…，a_r 的每一个性质. 存在这样的情况，其中我们想知道有多少个对象正好有这些性质中的 m 个性质. 对于 $m\leqslant r$，设 e_m 是正好有 m 个性质的对象数量. 为了表示 e_m 的公式，假设对于 $t\geqslant 1$，令

425

$$s_t=\sum N(a_{i_1}a_{i_2}\cdots a_{i_t}),$$

其中，上面的和是对 t 个不同性质 a_{i_1}，a_{i_2}，…，a_{i_t} 的所有选择求和. 于是，我们有下面的定理.

定理 7.4 *如果有 r 个性质且 $m\leqslant r$，那么正好有 m 个性质的对象的数量由下式给出：*

$$e_m = s_m - \binom{m+1}{1}s_{m+1} + \binom{m+2}{2}s_{m+2} - \binom{m+3}{3}s_{m+3} \pm \cdots$$
$$+ (-1)^p\binom{m+p}{p}s_{m+p} \pm \cdots + (-1)^{r-m}\binom{m+r-m}{r-m}s_r. \tag{7.15}$$

读者应该注意到，如果 s_0 取做 N，那么定理 7.4 中 $m=0$ 的特殊情况就是容斥原理.

我们将在 7.2.2 节证明这一定理. 这里，我们给出运用这一定理的几个例子. 特别地，我们返回到例 7.2. 有多少辆汽车正好超出一种污染物环境标准呢？我们要寻找 e_1. 为了计算 e_1，我们注意到，通过本例的计算，有

$$s_1 = 6+4+3 = 13,$$
$$s_2 = 3+2+1 = 6,$$
$$s_3 = 1.$$

因此，根据定理 7.4，有

$$e_1 = s_1 - \binom{2}{1}s_2 + \binom{3}{2}s_3$$
$$= 13 - 2(6) + 3(1)$$
$$= 4.$$

【例 7.8 衣帽存放问题(再探例 6.9)】 在例 6.9 中，我们考虑了这样的情况：n 位绅士的帽子被随机还回. 在这样的情况下，我们计算正好有一名绅士拿回自己帽子的方法数量. 考虑当帽子被随机还回时，帽子返还回各位绅士的可能方法的集合 A，这些方法对应于排列，设 a_i 是第 i 位绅士拿回自己帽子的性质. 在 7.1.7 节研究错位排列时，我们计算了下面的式子：

$$N(a_i)=(n-1)!,\quad \text{所有的 } i,$$
$$N(a_ia_j)=(n-2)!,\quad \text{所有的 } i\neq j,$$

而且一般地，有

426

$$N(a_{i_1}a_{i_2}\cdots a_{i_t})=(n-t)!.$$

因此，有

$$s_t=\binom{n}{t}(n-t)!$$

这是因为有$\binom{n}{t}$种方法选出 t 个性质 $a_{i_1}a_{i_2}\cdots a_{i_t}$. 于是，根据定理 7.4，对于 $r=n$ 和 $m=1$，我们有

$$\begin{aligned}
e_1&=s_1-\binom{2}{1}s_2+\binom{3}{2}s_3-\binom{4}{3}s_4\pm\cdots+(-1)^{n-1}\binom{n}{n-1}s_n\\
&=\binom{n}{1}(n-1)!-\binom{2}{1}\binom{n}{2}(n-2)!+\binom{3}{2}\binom{n}{3}(n-3)!-\binom{4}{3}\binom{n}{4}(n-4)!\\
&\quad\pm\cdots+(-1)^{n-1}\binom{n}{n-1}\binom{n}{n}(n-n)!\\
&=\frac{n!}{1!(n-1)!}(n-1)!-\frac{2!}{1!1!}\frac{n!}{2!(n-2)!}(n-2)!+\frac{3!}{2!1!}\frac{n!}{3!(n-3)!}(n-3)!\\
&\quad-\frac{4!}{3!1!}\frac{n!}{4!(n-4)!}(n-4)!\pm\cdots+(-1)^{n-1}\frac{n!}{(n-1)!1!}\\
&=n!-\frac{n!}{1!}+\frac{n!}{2!}-\frac{n!}{3!}\pm\cdots+(-1)^{n-1}\frac{n!}{(n-1)!}\\
&=n!\left[1-\frac{1}{1!}+\frac{1}{2!}-\frac{1}{3!}\pm\cdots+(-1)^{n-1}\frac{1}{(n-1)!}\right]\\
&=nD_{n-1}.
\end{aligned}$$

这一结果是显然的，因为我们只选出一名绅士拿回自己的帽子，这可以用 n 种方法完成，接着选出其余绅士的一个错位排列，这可以用 D_{n-1}种方法完成.

作为结论，正好有一名绅士拿回自己帽子的概率是

$$\frac{nD_{n-1}}{n!}=\frac{D_{n-1}}{(n-1)!},$$

当 n 趋近于无穷大时，这个值接近于 $1/e$，因为 $D_n/n!$ 趋近于 $1/e$. 因此，最后，正好有一位绅士拿回自己帽子的概率等同于没有绅士拿回自己帽子的概率. ■

【例 7.9 超能力测试】 在某些超能力实验中，我们按声称有超能力的人所不知的顺序给出一个 n 元素序列. 这个人事先预测该顺序. 我们计数正确元素数量，即那些正好处于预测序列位置上的元素的个数. 假设在一个 10 元素序列中，一个人得到 5 个正确元素. 我们能否以此作为超能力的证据呢？为了回答这个问题，我们要问，如果这个人只是猜测，那么观察到的成功数量是否是非常不可靠的. 特别地，我们问至少猜得 5 个正确元素的概率是多少(我们真正感兴趣的是这个人猜的至少如他所声称的那样的可能性是多少)？在 n 元素序列中正好猜中 m 个正确元素的

427

方法数量可以由定理 7.4 计算. 我们设 A 是集合$\{1, 2, \cdots, n\}$的所有排列的集合，且设 a_i 是 i 在第 i 个位置的性质. 于是 $N(a_i)$，$N(a_ia_j)$等等正好与我们在衣帽存放问题中所做的分析相同，对于每一个 t，s_t 也一样. 因此，可以证明，正好猜出 m 个正确位置的概率由下式给出：

$$P_m^n = \frac{1}{m!}\left[1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} \pm \cdots + (-1)^{(n-m)} \frac{1}{(n-m)!}\right]. \tag{7.16}$$

(7.16)式的详细验证留作练习(练习 26). 从 10 元素序列中至少猜出 5 个元素的正确位置的概率是:

$$P_5^{10} + P_6^{10} + P_7^{10} + P_8^{10} + P_9^{10} + P_{10}^{10} = 0.003\,06 + 0.000\,52 + 0.000\,07 + 0.000\,01 + 0.000\,00 + 0.000\,00 = 0.003\,66.$$

(注意，倒数第二项的 0.000 00 在这里事实上就是 0，而最后一项实际上是 1/10!). 我们的结论是，通过猜测得到这样大的成功概率非常小. 我们可以以此为证据得出这个人的确似乎有超能力的结论. 关于超能力测试进一步的参考资料以及错位排列在心理学的其他应用可以参见 Barton[1958]、Utts[1991]、Vernon[1936]. ■

【例 7.10 RNA 链】 我们来确定长度为 n 且正好有两个 U 的 RNA 链的数量. 我们可以直接计算这个值. 为了得到一个长度为 n 且正好有两个 U 的 RNA 链，我们可以从 n 个位置中选出两个位置给 U，然后对于其余 $n-2$ 个位置的每一个位置我们有三种基的选择. 这就给出了

$$\binom{n}{2}3^{n-2}$$

个链. 弄清楚如何从定理 7.4 得到这个结果是有意义的. 设 A 是字母表{U, A, C, G}上的所有长度为 n 的序列的集合，且设 a_i 是有一个 U 在第 i 个位置上的性质. 然后，我们寻找 e_2. 注意有下面的结果:

$$N(a_{i_1} a_{i_2} \cdots a_{i_t}) = 4^{n-t},$$

这是因为，如果 $i \neq i_1, i_2, \cdots, i_t$，则对于该链中的第 i 个元素我们有四种选择. 因此

$$s_t = \binom{n}{t}4^{n-t},$$

因为，我们是从 n 个性质中选出 t 个性质. 根据定理 7.4，我们得出结论:

$$\begin{aligned}
e_2 &= s_2 - \binom{3}{1}s_3 + \binom{4}{2}s_4 \mp \cdots + (-1)^p\binom{p+2}{p}s_{p+2} \pm \cdots + (-1)^{n-2}\binom{n}{n-2}s_n \\
&= \binom{n}{2}4^{n-2} - \binom{3}{1}\binom{n}{3}4^{n-3} + \binom{4}{2}\binom{n}{4}4^{n-4} \mp \cdots \\
&\quad + (-1)^p\binom{p+2}{p}\binom{n}{p+2}4^{n-p-2} \pm \cdots + (-1)^{n-2}\binom{n}{n-2}\binom{n}{n}4^{n-n}.
\end{aligned}$$

428

为了计算 e_2 的表达式，注意有下面的等式:

$$\begin{aligned}
(-1)^p\binom{p+2}{p}\binom{n}{p+2}4^{n-p-2} &= (-1)^p\frac{(p+2)!}{p!2!}\frac{n!}{(p+2)!(n-p-2)!}4^{n-p-2} \\
&= (-1)^p\frac{n(n-1)}{2}\frac{(n-2)!}{p!(n-p-2)!}4^{n-p-2} \\
&= \binom{n}{2}(-1)^p\binom{n-2}{p}4^{n-p-2}.
\end{aligned}$$

因此,

$$e_2 = \binom{n}{2}\sum_{p=0}^{n-2}\binom{n-2}{p}(-1)^p4^{n-p-2}.$$

根据二项式展开(定理 2.7)，有

$$e_2 = \binom{n}{2}(4-1)^{n-2} = \binom{n}{2}3^{n-2}.$$

这与我们最初的计算一致. 对于这一情况，使用定理 7.4 要困难得多. ■

【例 7.11 合法代码字(再探例 6.4)】 在例 6.4 中，我们定义了数字表{0，1，2，3}上的代码字是合法的，如果它有偶数个 0，而且我们设 a_k 是长度为 k 的合法代码字的数量. 在 6.3.1 节，我们使用生成函数证明了

$$a_k = \frac{1}{2}(2)^k + \frac{1}{2}(4)^k.$$

在此，我们将利用定理 7.4 推导出相同的结果. 设 A 是{0，1，2，3}上长度为 k 的所有序列的集合，且对于 $i=1$，2，…，$k=r$，设 a_i 是第 i 个数字是 0 的性质. 我们寻找有偶数个这些性质的 A 的元素的个数，即我们寻找 $e_0+e_2+e_4+\cdots$. 为了计算这个和，注意有

$$s_t = \binom{k}{t}4^{k-t}.$$

这是因为

$$N(a_{i_1}a_{i_2}\cdots a_{i_t}) = 4^{k-t}.$$

从这个等式和定理 7.4，通过代数变换可以证明：

$$e_0+e_2+e_4+\cdots = \frac{1}{2}(2)^k + \frac{1}{2}(4)^k. \tag{7.17}$$

429 证明(7.17)式的更简单方法是使用下面的定理，其证明将留到 7.2.2 节. ■

定理 7.5 *如果存在 r 个性质，有偶数个这样性质的对象的数量由下式给出：*

$$e_0+e_2+e_4+\cdots = \frac{1}{2}\left[s_0 + \sum_{t=0}^{r}(-2)^t s_t\right],$$

且有奇数个这样性质的对象的数量由下式给出：

$$e_1+e_3+e_5+\cdots = \frac{1}{2}\left[s_0 - \sum_{t=0}^{r}(-2)^t s_t\right].$$

把定理 7.5 运用于例 7.11，并回想到 s_0 被取作 N，我们就可发现，使用二项式展开(定理 2.7)有

$$\begin{aligned} e_0+e_2+e_4+\cdots &= \frac{1}{2}\left[4^k + \sum_{t=0}^{k}(-2)^t\binom{k}{t}4^{k-t}\right] \\ &= \frac{1}{2}[4^k + (-2+4)^k] \\ &= \frac{1}{2}[4^k + 2^k], \end{aligned}$$

这一结果与(7.17)式一致.

【例 7.12 宇宙射线和分装问题】 假设我们有一个暴露在宇宙射线下的带有 c 个盒子的盖林计数器，通过 n 条射线撞击它. 正好有 q 个计数器受到撞击的概率是多少? 为了回答这个问题，我们可以按着 7.1.4 节中的分析，引入样本空间 S 和事件 E_i. 这里，S 是由 n 条射线到 c 个盒子的所有分配组成的，而 E_i 是计数器 i 没有受到撞击的事件. 我们需要的是正好有 $m=c-q$ 个计数器没有受到撞击的概率，即正好有 m 个所讨论的事件发生的概率. 我们可以如 7.1.4 节那样引入一个集合 A 和性质 a_i，并观察到在诸事件 E_1，E_2，…，E_r 中，正好有 m 个事件发生的概率可以根据定理 7.4 并使用 $e_m/N(S)$ 计算. 在我们的例子中，我们可以把这一概率考虑成分装问题来计算 e_m，并使用 7.1.5 节中的 $N(a_i)$，$N(a_ia_j)$等的计算. 于是我们有

$$s_t = \sum N(a_{i_1}a_{i_2}\cdots a_{i_t}) = \binom{c}{t}(c-t)^n.$$

因此，根据定理7.4可以证明正好有m个事件E_1，E_2，…，E_r发生的概率由下式给出：

$$\binom{c}{m}\sum_{p=0}^{c-m}(-1)^p\binom{c-m}{p}\left(1-\frac{m+p}{c}\right)^n. \tag{7.18}$$

其详细验证留给读者(练习28). (7.18)式中的结果可以直接从$m=0$的情况推导出来(参见练习29). ■

对于定理7.4的其他应用，可以参见Feller[1968]、Irwin[1955]、Parzen[1992]. 430

7.2.2 定理7.4和定理7.5的证明[⊖]

我们通过给出定理7.4和定理7.5的证明来结束本节.

定理7.4的证明 这一证明类似于定理7.1的证明. 作为准备，我们注意到：

$$\binom{m+j}{m+p}\binom{m+p}{p}=\frac{(m+j)!}{(m+p)!(j-p)!}\frac{(m+p)!}{p!m!}=\frac{(m+j)!}{m!p!(j-p)!}$$
$$=\frac{(m+j)!}{m!j!}\frac{j!}{p!(j-p)!}=\binom{m+j}{m}\binom{j}{p}.$$

因此有

$$\binom{m+j}{m+p}\binom{m+p}{p}=\binom{m+j}{m}\binom{j}{p}. \tag{7.19}$$

现在，我们考虑等式(7.15). 如果一个对象有少于m个性质a_i，那么，e_m的计算不计数这一对象，而且(7.15)式的右边的任何项都不计数它. 假设一个对象正好有m个性质. e_m的计算正好计数它一次，而且(7.15)式的右边的计算正好计数它一次，即s_m的计算计数它一次. 最后，假设一个对象有多于m个性质，比如说有$m+j$个性质. e_m的计算不计数它. 我们将证明它在(7.15)式的右边的计数次数是0. 这个对象在计算s_m中的计数次数是$\binom{m+j}{m}$：对从对象所具有的$m+j$个性质中选取的每m个性质，它都被计数一次. 它在s_{m+1}的计算中的计数次数是$\binom{m+j}{m+1}$. 一般地，对于$p\leqslant j$，它在s_{m+p}的计算中的计数次数是$\binom{m+j}{m+p}$. 除此之外，它不被计数. 因此，这个对象在(7.15)式的右边的计数总次数的计算是：$\binom{m+j}{m+p}$乘以$(-1)^p\binom{m+p}{p}$，即s_{m+p}的系数，然后，对$p=0$到j把这些项相加. 因此，我们得到

$$\binom{m+j}{m}-\binom{m+j}{m+1}\binom{m+1}{1}+\binom{m+j}{m+2}\binom{m+2}{2}\mp\cdots$$
$$+(-1)^p\binom{m+j}{m+p}\binom{m+p}{p}\pm\cdots+(-1)^j\binom{m+j}{m+j}\binom{m+j}{j}. \tag{7.20}$$

现在，根据(7.19)式和(7.20)式，上式变成

431

$$\binom{m+j}{m}-\binom{m+j}{m}\binom{j}{1}+\binom{m+j}{m}\binom{j}{2}\mp\cdots+(-1)^j\binom{m+j}{m}\binom{j}{j},$$

这等于

$$\binom{m+j}{m}\left[\binom{j}{0}-\binom{j}{1}+\binom{j}{2}\mp\cdots+(-1)^j\binom{j}{j}\right]. \tag{7.21}$$

⊖ 本小节可以跳过.

根据定理 2.9，(7.21)式中括号内的结果等于 0[使用二项式展开把$(1-1)^j$ 展开就可以得到这一结果]，所以(7.21)式等于 0. 这就是定理(7.4)的证明. ■.

定理 7.5 的证明 设 $E(x)=\sum e_m x^m$ 是序列 e_0，e_1，e_2，…的普通生成函数. 根据定理 7.4 有：

$$
\begin{aligned}
E(x) &= [s_0 - s_1 + s_2 - \cdots + (-1)^r s_r] \\
&\quad + \left[s_1 - \binom{2}{1}s_2 + \binom{3}{2}s_3 - \cdots + (-1)^{r-1}\binom{r}{r-1}s_r\right]x \\
&\quad + \left[s_2 - \binom{3}{1}s_3 + \binom{4}{2}s_4 - \cdots + (-1)^{r-2}\binom{r}{r-2}s_r\right]x^2 + \cdots \\
&\quad + \left[s_m - \binom{m+1}{1}s_{m+1} + \binom{m+2}{2}s_{m+2} - \cdots + (-1)^{r-m}\binom{m+r-m}{r-m}s_r\right]x^m \\
&\quad + \cdots + s_r x^r \\
&= s_0 + s_1[x-1] + s_2\left[x^2 - \binom{2}{1}x + 1\right] + s_3\left[x^3 - \binom{3}{1}x^2 + \binom{3}{2}x - 1\right] + \cdots \\
&\quad + s_m\left[x^m - \binom{m}{1}x^{m-1} + \binom{m}{2}x^{m-2} + \cdots + (-1)^{m-1}\binom{m}{m-1}x + (-1)^m\right] + \cdots \\
&\quad + s_r\left[x^r - \binom{r}{1}x^{r-1} + \binom{r}{2}x^{r-2} + \cdots + (-1)^{r-1}\binom{r}{r-1}x + (-1)^r\right].
\end{aligned}
$$

432

因此，有

$$E(x) = \sum_{m=0}^{r} s_m (x-1)^m. \tag{7.22}$$

通过注意到有下面的等式：

$$e_0 + e_2 + e_4 + \cdots = \frac{1}{2}[E(1) + E(-1)],$$

并在(7.22)式中取 $x=1$ 和 $x=-1$，可以得到定理 7.5 的第一部分. 通过注意到下面的等式：

$$e_1 + e_3 + e_5 + \cdots = \frac{1}{2}[E(1) - E(-1)].$$

可以得到这一定理的第二部分. ■.

本节练习

1. 对于 7.1 节练习 1，正好预定其中一个频道的住户的百分比是多少?
2. 对于 7.1 节练习 2，多少个模块正好使用两个控制器?
3. 对于 7.1 节练习 4，多少人正好对其中的一种疾病有免疫力?
4. 对于 7.1 节练习 8，确定正好有所讨论的两种缺损的汽车的数量.
5. 对于 7.1 节练习 9，确定正好要参加三门所讨论课程考试的学生的数量.
6. 长度为 6 且有偶数个元音的单词的数量是多少?
7. 6.1.3 节中讨论的 Montmort 的"纸牌游戏问题"的一个变形如下所述. 把一副 n 张纸牌在桌子上排成一行. 第二副的 n 张纸牌一张一张随机地放置在第一副纸牌上. 如果第一副纸牌与第二副纸牌有 m 个匹配对，那么你得到 m 点.
   (a)如果 $n=4$，有多少种方法可以得到 2 点?
   (b)如果 $n=9$，得到 7 点的概率是多少?
8. 准备参加工作面试的 10 名不同求职者的卷宗上的名字不幸弄混了，一位新接待员把他们的名字随机地放置在卷宗上. 有多少种方法可以使得正好有 3 名求职者的卷宗得到正确的名字?
9. 在衣帽间问题中，使用 e_1 的公式确定，当有三名绅士时，正好有一名绅士拿回自己帽子的概率.

10. 在衣帽间问题中，如果有4位绅士，计算这些绅士中正好有两名绅士拿回自己的帽子的方法数.

11. 对于任意的m，计算衣帽间问题中的e_m.

12. (a)当投掷4枚硬币时，使用定理7.4计算正好有2个正面的概率. 433

(b)通过直接计算检验答案.

13. 当1枚硬币被投掷n次时，使用定理7.4计算正好得到m个正面的方法数量.

14. (a)使用定理7.4确定{1，2，3，4，5，6，7，8}的排列中正好有4个数字在其自然位置上的排列数量.

(b)通过直接计算检验答案.

15. (a)假设数字表{0，1，2}上的代码字是合法的，当且仅当它正好有3个1. 使用定理7.4确定长度为7的合法代码字的数量.

(b)用另一种计算检验答案.

16. (a)假设数字表{0，1，2}上的代码字是合法的，当且仅当它正好有5个1. 使用定理7.4计算长度为n的合法代码字的数量.

(b)用另一种计算检验答案.

17. (a)假设一个家庭出生了n个孩子. 使用定理7.4计算这个家庭正好有两个男孩的方法数量.

(b)用第2章的方法检验答案.

18. (a)一位超能力者预测4个元素的序列，得到2个正确结果. 至少得到2个正确结果的概率是多少?

(b)得到3个或更多正确结果的概率是多少?

(c)正好得到3个正确结果的概率是多少?(你的答案有问题吗? 请给予解释.)

19. 在一次品酒实验中，一名验酒员被告知将给他5种不同的酒. 在每次品尝之后，他猜测这5种酒都是什么，而且确定没有做重复的猜测. 他猜对3种. 如果他做随机的猜测，那么他至少猜对3种酒的概率是多少?

20. 写出在从0，1，2，…，9随机选出的7个数字组成的序列中，正好有2个数字不出现的概率的表达式.

21. 在一个遗传学实验中，一窝n只老鼠中的每一只老鼠被分类属于m个基因类型中的一个. 在这n只老鼠中正好有g个基因类型的概率是多少?

22. 考虑有10个孩子的家庭. 使用定理7.5确定有偶数个男孩的家庭的数量. 通过直接计算检验答案.

23. 考虑数字表{0，1，2}上的8数字序列，使用定理7.5确定有奇数个1的序列的个数. 通过直接计算检验答案.

24. 确定长度为8且没有基U而且正好有偶数个G的RNA链的数量.

25. 通过利用数学归纳法给出定理7.4的另一个证明.

26. 使用定理7.4验证等式(7.16).

27. 对于7.1节练习27，证明放置CD使得正好有m个播放器为空的方法数量由下式给出： 434

$$\binom{n}{m}\sum_{i=0}^{n-m}(-1)^i\binom{n-m}{i}(n-m-i)^d.$$

28. 使用定理7.4验证(7.18).

29. 如果把n个可区分球分配到c个可区分盒子. 假设$P_m(c,n)$是正好有m个盒子为空的概率.

(a)证明：

$$P_m(c,n)=\binom{c}{m}\left(1-\frac{m}{c}\right)^n P_0(c-m,n).$$

(b)通过$P_0(c-m,n)$的等式推导(7.18)式.

30. 设e_m^*是集合A中至少有性质a_1，a_2，…，a_r中m个性质的元素的个数. 证明：

$$e_m^*=s_m-\binom{m}{m-1}s_{m+1}+\binom{m+1}{m-1}s_{m+2}\mp\cdots+(-1)^p\binom{m+p-1}{m-1}s_{m+p}\pm\cdots$$
$$+(-1)^{r-m}\binom{m+r-m-1}{m-1}s_r.$$

31. 假设 E_1，E_2，…，E_r 是事件，且 $p_{i_1 i_2 \cdots i_t}$ 是事件 E_{i_1}，E_{i_2}，…，E_{i_t} 全部发生的概率，而且假设 $S_t = \sum p_{i_1 i_2 \cdots i_t}$，其中的和是对{1，2，…，$r$}的所有 t 子集{i_1，i_2，…，i_t}的求和. 利用 S_t，推导下列表达式：
    (a)正好有 m 个事件发生的概率的表达式.
    (b)至少有 m 个事件发生的概率的表达式.
32. 对于 7.1 节练习 8，至少有 2 个所讨论缺损的汽车有多少？
33. 对于 7.1 节练习 9，至少要接受所讨论的课目之一考核的学生有多少？
34. 计算长度为 10 且至少有 2 个 U 的 RNA 链的数量.
35. 假设数字表{0，1，2}上的代码字是合法的，当且仅当它至少有 3 个 1. 计算长度为 7 的合法代码字的数量.
36. 计算{1，2，3，4，5}的排列中至少有 3 个整数在其自然位置上的排列的数量.
37. 假设 B′是 $n \times m$ 棋盘 B 的补，且 $n \leqslant m$. 如果 $j \leqslant n$，确定使用 $r_k(B')$表示的 $r_j(B)$的公式，这一结果扩展了等式(7.11)的结果.
38. 使用练习 37 的结果证明：

$$R(x,B) = x^n R\left(\frac{1}{x}, B'\right).$$

39. (a)如果 $E(x)$是序列 e_0，e_1，e_2，…的普通生成函数，且 e_i 如例 7.11 中的定义，那么 $E(1)$是什么？
    (b)确定在一般情况下成立的 $E(1)$的公式.

435

参考文献

ADLEMAN, L. M., and HUANG, M.-D., *Primality Testing and Two Dimensional Abelian Varieties over Finite Fields*, Lecture Notes in Mathematics, 1512, Springer-Verlag, Berlin, 1992.

ADLEMAN, L. M., POMERANCE, C., and RUMELY, R. S., "On Distinguishing Prime Numbers from Composite Numbers, *Ann. Math., 117*, (1983), 173–206.

AGRAWAL, M., KAYAL, N., and SAXENA, N., "PRIMES Is in P," Preprint, Aug. 6, 2002. http://www.cse.iitk.ac.in/primality.pdf

ANSHEL, I., ANSHEL, M., and GOLDFELD, D., "An Algebraic Method for Public-Key Cryptography," *Math. Res. Lett., 6* (1999), 287–291.

BARTON, D. E., "The Matching Distributions: Poisson Limiting Forms and Derived Methods of Approximation," *J. Roy. Statist. Soc., 20* (1958), 73–92.

BERGE, C., *Principles of Combinatorics*, Academic Press, New York, 1971.

BIRKHOFF, G. D., "A Determinant Formula for the Number of Ways of Coloring a Map," *Ann. Math., 14* (1912), 42–46.

COHEN, D. I. A., *Basic Techniques of Combinatorial Theory*, Wiley, New York, 1978.

DE MOIVRE, A., *The Doctrine of Chances*, private printing, London, 1718.

DIFFIE, W., and HELLMAN, M. E., "New Directions in Cryptography," *IEEE Trans. Info. Theory, 22* (1976), 644–654.

FELLER, W., *An Introduction to Probability Theory and Its Applications*, 3rd ed., Wiley, New York, 1968.

GOLDSTEIN, L. J., SCHNEIDER, D. I., and SIEGEL, M. J., *Finite Mathematics and Its Applications*, 7th ed., Prentice Hall, Upper Saddle River, NJ, 2001.

GOLDWASSER, S., and KILIAN, J., "Almost All Primes Can Be Quickly Certified," Proceedings of Annual ACM Symposium on Theory of Computing, 1986, 316–329.

HALL, M., *Combinatorial Theory*, 2nd ed., Wiley, New York, 1986.

HARDY, G. H., and WRIGHT, E. M., *Introduction to the Theory of Numbers*, 5th ed., Oxford University Press, New York, 1980.

IRWIN, J. O., "A Unified Derivation of Some Well-Known Frequency Distributions of Interest in Biometry and Statistics," *J. Roy. Statist. Soc., Ser. A, 118* (1955), 389–404.

KOBLITZ, N., *A Course in Number Theory and Cryptography*, 2nd ed., Springer-Verlag, New York, 1994.

LIU, C. L., *Topics in Combinatorial Mathematics*, Mathematical Association of America, Washington, DC, 1972.

MILLER, G. L., "Riemann's Hypothesis and Tests for Primality," *J. Comput. Sys. Sci., 13* (1976), 300–317.

MIZRAHI, A., and SULLIVAN, M., *Finite Mathematics; An Applied Approach*, Wiley, New York, 1999.

PARZEN, E., *Modern Probability Theory and Its Applications*, Wiley, New York, 1992.

RABIN, M. O., "Probabilitic Algorithm for Testing Primality," *J. Number Theory, 12* (1980), 128–138.

REINGOLD, E. M., NIEVERGELT, J., and DEO, N., *Combinatorial Algorithms: Theory and Practice*, Prentice Hall, Englewood Cliffs, NJ, 1977.

RHEE, M. Y., *Cryptography and Secure Communications*, McGraw-Hill, New York, 1993.

RIVEST, R. L., SHAMIR, A., and ADLEMAN, L. M., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Comm. ACM, 21* (1978), 120–126. (See also U.S. Patent 4,405,829, 1983.)

ROTA, G. C., "On the Foundations of Combinatorial Theory. I. Theory of Möbius Functions," *Z. Wahrscheinlichkeitsth. Verw. Geb., 2* (1964), 340–368.

436

SCHNEIER, B., *Applied Cryptography: Protocols, Algorithms and Source Code in C*, Wiley, New York, 1995.

SEBERRY, J., and PIEPRZYK, J., *Cryptography: An Introduction to Computer Security*, Prentice Hall, Englewood Cliffs, NJ, 1989.

SHEN, A., "Probabilistic Proofs," *The Mathematical Intelligencer, 20* (1998), 29–31.

SHOR, P. W., "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM J. Computing, 26* (1997), 1484–1509.

SOLOVAY, R., and STRASSEN, V., "A Fast Monte-Carlo Test for Primality," *SIAM J. Computing, 6* (1977), 84–85.

STALLINGS, W., *Cryptography and Network Security: Principles and Practice*, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 1999.

UTTS, J. M., "Replication and Meta-Analysis in Parapsychology," *Statistical Science, 6* (1991), 363–403.

VERNON, P. E., "The Matching Method Applied to Investigations of Personality," *Psychol. Bull., 33* (1936), 149–177.

VILENKIN, N. YA., *Combinatorics*, Academic Press, New York, 1971. (Translated from the Russian by A. Shenitzer and S. Shenitzer.)

WHITNEY, H., "A Logical Expansion in Mathematics," *Bull. Amer. Math. Soc., 38* (1932), 572–579.

437

第 8 章　波利亚计数理论[⊖]

8.1　等价关系

8.1.1　不同的构形与数据库

在本书中，我们已讨论了各类构形的计数．在科学的各个领域以及科学应用的各个领域中，越来越多的各类构形应用于巨型数据库中．这些构形可能非常复杂．在医学决策系统中，有医疗图像的大型数据库．在分子生物学研究中，有蛋白质结构的巨型数据库．在环境建模中，有环境特性的巨大数据库．电信和信用卡公司有通信和消费模式的巨大数据库来帮助发现欺诈行为．存储于这些巨型数据内的构形通常是复杂的几何对象，或带有各种维数或性质的对象．这些巨型数据库使得搜索、恢复甚至组织等成为令人生畏的问题．有时候，计数特定类型构形的数量以便帮助估测数据库中的搜索长度是很有用的．我们遇到的问题之一是要确定两个构形是否相同．本章开发若干用于计数特定类型的不同构形数量的技术．当然，这些技术的充分利用涉及精确地确定两个构形是否相等的概念．因此，在本章的开始，我们研究两个构形相同的含义．与之前的例子相比，本章的例子能够使“相同”的概念更加清晰．我们将考虑有机分子、着色树、开关函数、弱序等中的相同概念．因为目前组合数学在更新但没有精确的定义的场合变得越来越重要，所以本章所描述的方法对这样的场合会很有用．

439

8.1.2　等价关系的定义

假设 V 是集合，且 S 是 V 的元素的序对的集合．在 4.1.1 节，我们称 S 是 V 上的(二元)关系(binary relation)．例如，如果 $V=\{1, 2, 3\}$ 且 $S=\{(1, 2), (2, 3)\}$，那么 S 是 V 上的一个关系．如果序对 (a, b) 在 S 内，则我们写作 aSb．因此，在我们的例子中，$1S2$ 成立，但 $2S1$ 和 $1S3$ 不成立．

假设 V 是一个构形的集合，且对于 V 中的元素 a 和 b，我们写 aSb 意味着 a 和 b 相同．那么关系 S 将有下面的性质(在 4.1.2 节已定义)：

自反性(reflexivity)：对于 V 中所有的 a，有 aSa．(任意构形都与自身相同．)

对称性(symmetry)：对于 V 中所有的 a，b，如果有 aSb，那么有 bSa．(如果 a 与 b 相同，那么 b 与 a 相同．)

传递性(transitivity)：对于 V 中所有的 a，b，c，如果 aSb 且 bSc，那么 aSc．(如果 a 与 b 相同，且 b 与 c 相同，那么 a 与 c 相同．)

如果 S 满足这三个性质，则称它为一个等价关系(equivalence relation)．

现在，我们给出若干其他等价关系的例子．设 V 是新泽西人的集合，且设 aSb 意味着 a 和 b 的身高相同．那么 S 定义一个等价关系．设 V 是所有美国人的集合，且设 aSb 意味着 a 和 b 有相同的生日．那么 S 是等价关系．设 V 是所有美国人的集合，且设 aSb 意味着 a 是 b 的父亲．那么 S 不定义等价关系：它既不是自反的，也不是对称的，也不是传递的．设 V 是实数集合，且设 aSb 意味着 $a\leqslant b$．那么 S 不是等价关系：它是自反的，也是传递的但是它不是对称的．

我们给出若干更加复杂的例子来说明本章要讨论的基本问题．

⊖　作为基础课程，本章应该跳过．在很多地方，本章需要代数的支持．

【例 8.1 着色 2×2 阵列】 让我们考虑一个 2×2 的阵列，其中每一个块或者被占领或不被占领. 当一个块被占领时，我们把它涂成深色，否则我们把它涂成浅色或不涂色. 图 8.1 给出若干这样的着色. 设 V 是所有这样的着色的全体. 假设允许这个阵列旋转[⊖] 0°、90°、180°，或者 270°. 我们认为被着色的阵列 a 和 b 相同，且写作 aSb，如果 b 能够从 a 通过旋转上述的某个角度而得到. 那么 S 定义一个等价关系. 为了说明，注意图 8.1 给出了关系 S 定义的一些阵列对. 为了弄清为什么 S 是一个等价关系，注意它是自反的，因为 a 可以从 a 通过转旋 0°而得到. 它又是对称的，因为如果 b 可以从 a 通过旋转 x 度而得到，那么可以从 b 通过旋转 $360-x$ 度得到 a. 最后，它是传递的，因为如果 b 可以从 a 通过旋转某个角度而得到，且 c 从 b 通过旋转另一个角度而得到，那么 c 可以从 a 通过第一个旋转，再通过第二个旋转而得到(这里假设旋转 $360+x$ 度等价于旋转 x 度). ■

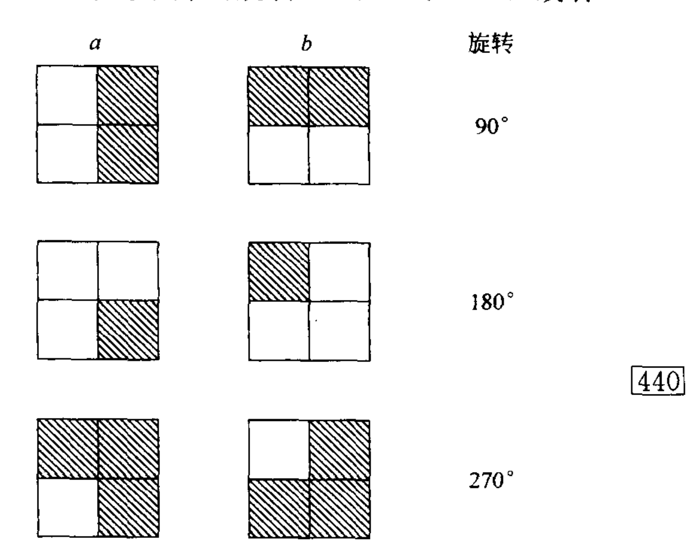


图 8.1 2×2 阵列的着色. 对于每一种情况，因为 b 可以从 a 通过一个旋转而得到，所以有 aSb

440

【例 8.2 项链】 假设一条开式项链是由一串 k 个珠子组成的，且每个珠子不是蓝的就是红的. 因此，一条三个珠子的典型项链可以表示成诸如 bbr 或 brb 等的位串. 我们认为一条项链没有指定的首尾，所以两条这样的项链 x 和 y 被认为相同，即我们写作 xSy，如果 x 等于 y 或 y 可以通过将 x 返转过来得到. 因此 bbr 与 rbb 相同. S 定义一个等价关系. 这一事实的证明留给读者(练习 4). ■

【例 8.3 开关函数(再探例 2.4)】 回想一下例 2.4，n 变量开关函数是给每一个长度为 n 的位串指定一个数 0 或者 1 的函数. 在计算机工程学中会遇到这样的函数. 根据我们在例 2.4 中所进行的讨论，某些开关函数被认为是等价的或相同的. 为了使这一意义更精确，假设 T 和 U 是由表 8.1 定义的两个开关函数. 很容易看到对于所有的位串 x_1x_2，$T(x_1x_2)=U(x_2x_1)$. 因此，T 可以从 U 通过简单地改变输入顺序，即交换两个输入的位置而得到. 在这一意义下，T 和 U 可以认为是等价的. 实际上，对于所有实际用途，它们是等价的. 因为，假设我们可以设计一个计算 U 的电子回路. 那么，我们就可以如图 8.2 所示那样把计算 U 的电子回路看成一个黑盒子(不需要知道具体的电子回路)设计计算 T 的电子回路.

表 8.1 两个开关函数 T 和 U

位串 x	$T(x)$	$U(x)$
00	1	1
01	0	1
10	1	0
11	1	1

441

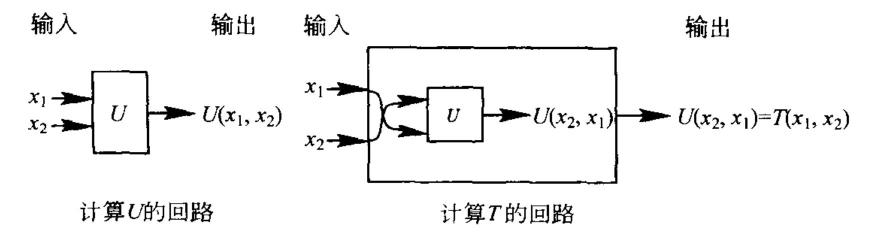


图 8.2 计算 T 的电子回路可以从计算 U 的回路得到

⊖ 除非特别说明，否则本章中的所有旋转都是按逆时针的.

一般地，我们认为两个二变量开关函数 T 和 U 相同，并写作 TSU，如果或者 $T=U$，或者对于所有位串 x_1x_2 有 $T(x_1x_2)=U(x_2x_1)$. 这时，S 是一个等价关系. 我们把这一事实的证明留给读者(练习 5). 在下文中，我们把等价的概念扩展到含两个以上变量的开关函数. 在 2.1 节我们已经注意到，即使是 4 个变量，也有很多开关函数. 因此，为每一个 n 变量开关函数编译一个完全对应于电子回路的手工列表是不切实际的. 然而，没有必要在这样的列表中包含每一个开关函数，而只需包含足够多的开关函数使得每一个 n 变量开关函数等价于列表中的某个开关函数即可. 计数所需的开关函数的数量历来是计算机科学中的一个重要问题(参见 2.1 节)，我们将展示如何进行这一计算. ■

【例 8.4 着色树】㊀ 设 T 是一棵固定的树，例如，如图 8.3 所示的 7 顶点二叉树. 把 T 的每一个顶点着色成黑色或白色，但不做左右之分. 设 V 是 T 的所有着色的全体. 设 aSb 意味着 a 和 b 被认为是相同的，即 b 可以从 a 通过交换左右子树而得到. 后面我们将对此给出更精确的定义. 然而，因为我们不区分左右子树，所以显然图 8.3 中的所有着色都是相同的. 根据我们下面的一般结果，S 定义一个等价关系.

442

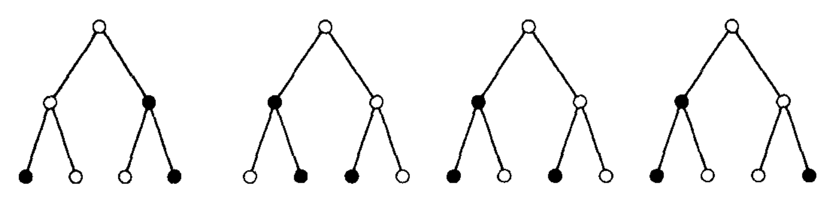


图 8.3　7 顶点二叉树的四个等价着色

■

【例 8.5 有机分子】㊁ 本章阐述的理论的重要历史动机之一是希望计数化学中不同有机分子的数量. 考虑如图 8.4 所示形式的分子的集合 V，其中 C 是碳原子，每一个 X 是 CH_3(甲基)、C_2H_5(乙醛)、H(氢)和 Cl(氯)之一. 一个这样的典型分子是 CH_2Cl_2，这一分子有两个氢原子和两个氯原子. 我们可以用正四面体模型化这样的分子，正四面体由相交于六条边和四个角的等边三角形组成的，如图 8.5 所示. 我们可以把这个碳原子想象为处于正四面体的中心，而标签为 X 的四个成分处于标签分别为 a，b，c 和 d 的四个角上. 两个这样的分子被认为相同，写作 xSy，如果 y 能够从 x 通过下列的四面体的 12 种对称变换而得到：没有变化；围绕连接一个顶点与其对立面中心点的直线旋转 120°或者 240°(存在 8 个这样的旋转)；围绕连接对边中点的直线旋转 180°(存在三种这样的旋转). 图 8.6 和图 8.7 分别说明了第二种和第三种对称变换.

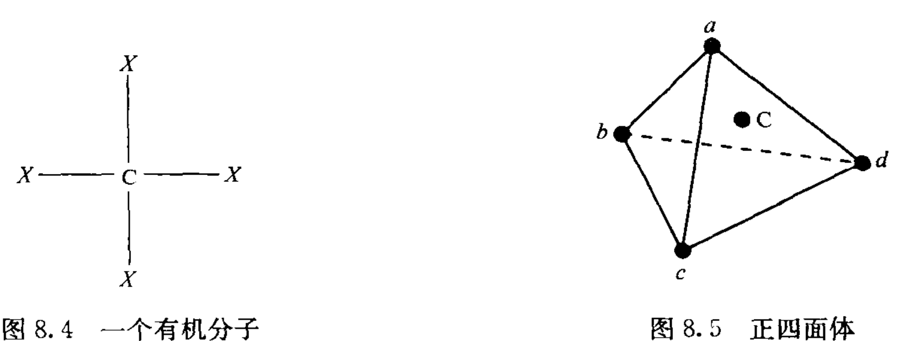


图 8.4　一个有机分子　　　　图 8.5　正四面体

㊀ 本例子取自 Reingold，Nievergelt 和 Deo[1977].

㊁ 本例取自 Liu[1968]. 根据本章的观点对化合物的更广泛的论述参见 Pólya 和 Read[1987].

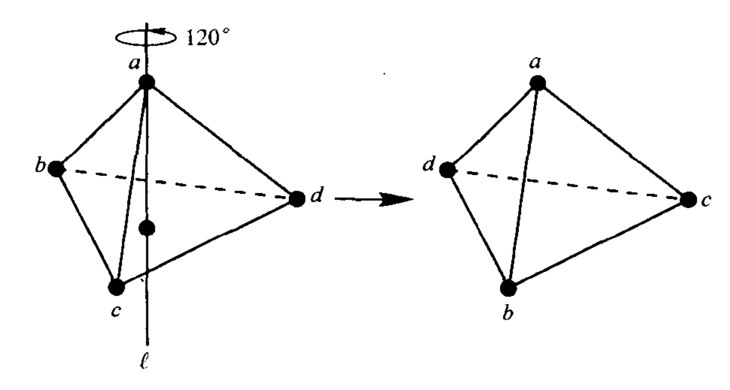


图 8.6 围绕连接顶点 a 和其对立面中心的直线 l 旋转 120°

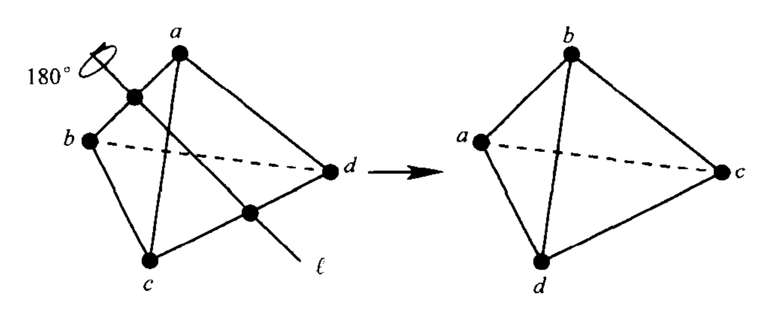


图 8.7 围绕连接对边中点的直线 l 旋转 180°

■

【例 8.6 弱序数量(再探例 5.29)】 回想一下 4.2 节的内容，图 4.14 给出了集合 A 上的一个典型的弱序 R. 每一个元素都有一个水平层次，在相同水平层次上的所有元素 a 和 b 满足 aRb 和 bRa，否则，aRb 当且仅当 a 所处的层次高于 b. 我们认为集合 A 上的两个弱序是相同的，如果它们的层次数量相同，且处于对应层次上的元素数量相同. 例如，图 8.8 所示的前两个弱序是相同的. 第一和第三个弱序作为弱序事实上是恒等的，因为它们有相同的序对集合，$\{(a, c), (a, d), (a, e), (a, f), (a, g), (a, h), (b, c), (b, d), (b, e), (b, f), (b, g), (b, h), (c, d), (c, e), (c, f), (c, g), (c, h), (d, g), (d, h), (e, g), (e, h), (f, g), (f, h)\}$. 容易看到，如果 aSb 意味着 a 和 b 是相同的，那么 S 定义一个弱序间的一个等价关系. 图 8.9 给出 $\{1, 2, 3\}$ 上的所有可能的弱序. 例如，注意有 ASD，GSI，JSL.

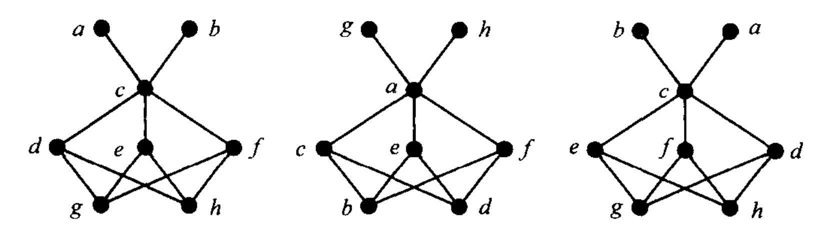


图 8.8 相同的三个弱序

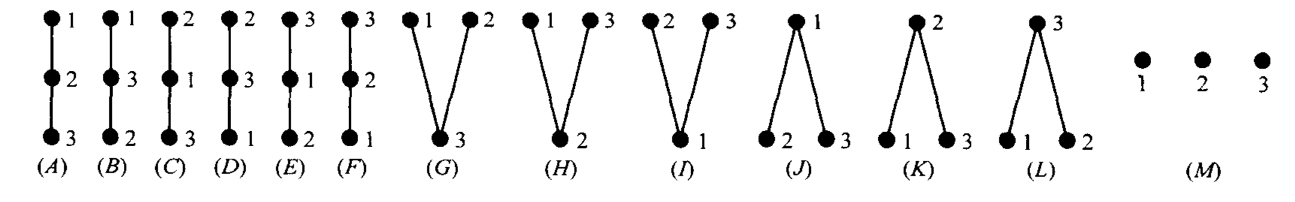


图 8.9 $\{1, 2, 3\}$的所有弱序

■

8.1.3　等价类

集合 V 上的等价关系 S 把 V 的元素分成称之为等价类(equivalence classes)的类. 特别地，如果 a 是 V 中的任意元素，包含 a 的等价类(equivaalence class containing)，$C(a)$是由所有满足 aSb 的元素 b 组成的，即 $C(a)=\{b\in V: aSb\}$. 根据自反性，aSa，所以 V 中的每一个元素都在某个等价类中；特别地，$a\in C(a)$. 另外，对于 V 中所有的 a，b，或者 $C(a)=C(b)$或者 $C(a)$和 $C(b)$互不相交. 因为假设 x 既在 $C(a)$中又在 $C(b)$中，那么有 aSx 和 bSx 成立. 根据对称性，有 aSx 和 xSb 成立. 传递性蕴涵 aSb. 这表明 $C(a)=C(b)$. 因为如果 y 在 $C(b)$中，那么 bSy 成立. 现在，aSb 和 bSy 蕴涵 aSy 成立，所以 y 在$C(a)$中. 因此有 $C(b)$包含于 $C(a)$. 类似地，我们可以证明 $C(a)$包含于 $C(b)$. 因此，$C(a)=C(b)$. 现在，如果我们把 $C(a)$和 $C(b)$看成是相等的，如果它们有相同的元素，那么我们有下面的定理.

443～445

定理 8.1　*如果 S 是等价关系，那么每一个元素在且只在一个等价类中.*

为了说明这一结果，注意在例 8.2 中，如果项链的长度为 2，则有四种项链，bb，br，rb 和 rr. 第二个和第三个是等价的，因此，$C(bb)=\{bb\}$，$C(br)=\{br, rb\}$. 存在三种不同的等价类，$\{bb\}$，$\{br, rb\}$和$\{rr\}$.

在例 8.1 中，存在 6 个等价类. 如图 8.10 所示.

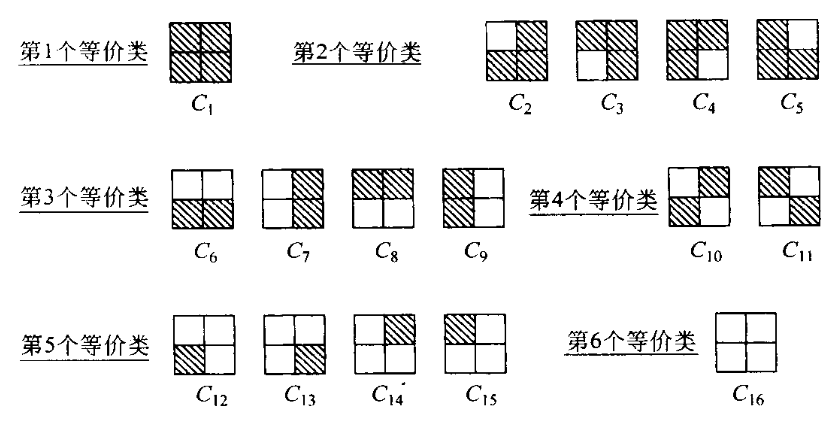


图 8.10　黑白着色的 2×2 阵列的等价类

本节练习

1. 对于下列每一种情况，S 是 V 上的等价关系吗？如果不是，确定它有哪些等价关系的性质？

(a)V 是实数，aSb 当且仅当 $a=b$.

(b)V 是实数，aSb 当且仅当 $a\neq b$

(c)V 是实数，aSb 当且仅当 b 除以 a 为整除.

(d)$V=\{1, 2, \cdots, n\}$的所有子集，aSb 当且仅当 a 和 b 的元素数量相同.

(e)V 同(d)中的定义，aSb 当且仅当 a 和 b 重叠.

(f)$V=$全世界人民，aSb 当且仅当 a 是 b 的兄弟.

(g)$V=$全美国人，aSb 当且仅当 a 和 b 有相同的血型.

446

(h)$V=\{1, 2, 3, 4\}$，$S=\{(1, 1), (2, 2), (3, 4), (4, 3), (1, 3), (3, 1)\}$.

(i)$V=\{w, x, y, z\}$，$S=\{(x, x), (y, y), (z, z), (w, w), (x, z), (z, x), (x, w), (w, x), (z, w), (w, z)\}$.

(j)$V=$所有加利福尼亚的居民，aSb 当且仅当 a 和 b 的居住地相距不超过 10 英里.

2. 假设 V 是长度为 4 的位串的集合，aSb 成立当且仅当 a 和 b 中 1 的数量相同. (V, S)是等价关系吗?
3. 对下列每一个等价关系，确定所有的等价类.
   (a)$V=\{a, b, c, d\}$，$S=\{(a, a), (b, b), (c, c), (d, d), (a, b), (b, a), (c, d), (d, c)\}$.
   (b)$V=\{u, v, w\}$，$S-\{(u, u), (v, v), (w, w), (u, v), (v, u), (v, w), (w, v), (u, w), (w, u)\}$.
   (c)$V=\{1, 2, 3, 4\}$，$S=\{(1, 1), (2, 2), (3, 3), (4, 4), (1, 2), (2, 1), (2, 3), (3, 2), (1, 3), (3, 1)\}$.
   (d)$V=$所有正整数的集合，aSb 当且仅当 $|a-b|$ 是偶数
   (e)$V=\{1, 2, 3, \cdots, 15\}$，$aSb$ 当且仅当 $a\equiv b$(模 3).
4. 证明例 8.2 中的 S 是等价关系.
5. 证明例 8.3 中的 S 是二变量开关函数间的等价关系.
6. 对于例 8.2，确定长度为 3 的项链的等价类.
7. 对于例 8.2，当每个珠子是三种颜色蓝、红或紫色中的一个时，确定长度为 2 的项链的等价类.
8. 对于例 8.1，假设我们可以使用三种颜色中的任意一种：黑色(b)、白色(w)或红色(r). 描述着色的所有等价类.
9. 对于例 8.1，假设我们不仅允许旋转，而且还允许在垂直、水平或对角线方向上的反射(对角线方向上的反射即交换对角线两侧的格子的指定颜色). 确定着色的所有等价类(仅使用两种颜色：黑色或白色).
10. 对于例 8.6，确定图 8.9 中的弱序的等价类.
11. 对于例 8.6，描述$\{1, 2, 3, 4\}$上的弱序的等价类.
12. 对于例 8.3，确定二变量开关函数的所有等价类.
13. 对于图 8.11 中的每棵树，画出在例 8.4 的意义下的与其等价的所有树.
14. 一个位串 x 的补 x' 是从 x 通过互换所有 0 和 1 而得到的. 例如，如果 $x=00110$，那么 $x'=11001$. 假设我们认为两个 n 变量开关函数 T 和 U 相同，如果 $T=U$ 或者对于所有的位串 x 有 $T(x)=U(x')$. 如果 $n=3$，描述在此相等关系的意义下开关函数的所有等价类.
15. 假设 V 是图 8.12 中的二叉树的全部着色的集合，图中每一个顶点得到黑色或白色. 如果两个着色认为是相等的，如果其中的一种着色可以从另一种着色通过将标有 1 和 2 的顶点的着色互换而得到. 确定在这一意义下的所有着色等价类.

447

a)　　b)

图 8.11　8.1 节练习 13 的树

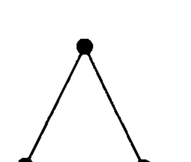


图 8.12　8.1 节练习 15 的树

16. 对于例 8.5，假设一个分子有三个氢原子和一个非氢原子. 有多少其他带有三个氢原子和一个非氢原子的分子被认为与这个分子相同?
17. 考虑一个正方形，并设 V 是使用红和蓝两种颜色着色它的顶点时的全部着色的集合. 对于着色 f 和 g，令 fSg 成立，如果 g 可以从 f 通过将正方形旋转 0°、90°、180°或 270°而得到. 证明(V, S)是一个等价关系，并确定所有的等价类.
18. 扩展练习 17，设 V 是使用颜色集合$\{1, 2, \cdots, n\}$中的颜色着色一个正 p 边形的顶点时的所有着色的集合，对于着色 f 和 g，设 fSg 成立，如果 g 可以从 f 通过把这个 p 边形旋转 $k(360/p)$角度中的一个而得到，其中 $k=0, 1, \cdots, p-1$. 对于下列条件计数等价类的数量：
   (a)$p=5$，$n=2$　　(b)$p=6$，$n=3$　　(c)$p=12$，$n=2$

19. 重复练习 17，其中 fSg 成立，如果 g 可以通过下列方法从 f 得到：把这个正方形旋转 0°、90°、180°或 270°，或者，通过这个正方形对角连线的反射，或通过这个正方形对边中点连线的反射.

20. 考虑集合 $A=\{1, 2, \cdots, n\}$.

  (a) A 上可能的二元关系有多少?

  (b) A 上可能的自反关系有多少?

  (c) A 上可能的对称关系有多少?

448

  (d) A 上可能的传递关系有多少?

  (e)当 $n=4$ 时，A 上可能的等价关系有多少?

21. 假设 V 是 n 顶点无标签图的集合，且 aSb 当且仅当 a 和 b 是同构的.

  (a)证明 S 是 V 上的等价关系.

  (b)如果 $n=3$，根据每一个等价类确定一个无标签图.

22. 设 E_n 等于集合 $A=\{1, 2, \cdots, n\}$ 上的等价关系的数量. 证明 E_n 满足下面的递推关系：

$$E_n = \sum_{i=0}^{n-1} \binom{n-1}{i} E_i, \quad n \geqslant 1.$$

8.2 排列群[⊖]

8.2.1 排列群的定义

在诸如例 8.1～例 8.5 的研究中，我们着重利用排列的概念. 回想一下，集合 $A=\{1, 2, \cdots, n\}$ 上的一个排列是 A 中元素的一个排序. 1 传递到 a_1，2 传递到 a_2 等的排列可以写成如下形式：

$$\begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ a_1 & a_2 & a_3 & \cdots & a_n \end{pmatrix},$$

或简写成 $a_1a_2\cdots a_n$. 因此，排列 132 可以表示成

$$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}.$$

类似地，排列

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix}$$

可以写成 3142.

集合 A 的排列也可以看成是 A 到其自身上的一个函数. 这个函数一定是一一对应的. 因此，排列

$$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}$$

可以看成是这样的函数 π：$\{1, 2, 3\} \to \{1, 2, 3\}$，其中 $\pi(1)=1$，$\pi(2)=3$，$\pi(3)=2$. 类似地，如果 A 是任意有限集合，任意从 A 到 A 的一一对应函数都可以看成是 A 上的排列；我们使用整数 1，2，…，n 表示 A 的元素. 例如，假设 $A=\{a, b, c, d\}$ 和 $f(a)=b$，$f(b)=c$，$f(c)=d$，$f(d)=a$. 如果 $a=1$，$b=2$，$c=3$，$d=4$，则 f 可以看成是这样的排列：

449

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}.$$

⊖ 在数学中，排列群更多译作置换群. 这里，一是考虑术语的一致性，二是考虑排列的操作意思更强一些，更适合对应用感兴趣的读者. ——译者注

假设 π_1 和 π_2 是集合 A 的排列. 我们可以定义 π_1 和 π_2 的积(product)或合成(composition) $\pi_1 \circ \pi_2$ 为这样的排列：首先通过排列 π_2 进行排序，然后再通过排列 π_1 对上述排列结果再进行排序. 例如，如果

$$\pi_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 1 & 3 \end{pmatrix} \quad 和 \quad \pi_2 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix},$$

那么

$$\pi_1 \circ \pi_2 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 1 \end{pmatrix}.$$

因为π_2 把 1 传递到 2，而π_1 把 2 传递到 2，所以π_1 和π_2 的合成把 1 传递到 2. 即 $\pi_1 \circ \pi_2(1) = \pi_1(\pi_2(1)) = \pi_1(2) = 2$. 类似地，$\pi_2$ 把 2 传递到 1，而 π_1 把 1 传递到 4，所以它们的合成把 2 传递到 4；以此类推.

设 X 是集合 A 的所有排列的集合. 注意，这一排列的集合满足下面的条件：

条件 **G1**(封闭性，closure). 如果 $\pi_1 \in X$ 且 $\pi_2 \in X$，那么 $\pi_1 \circ \pi_2 \in X$.

条件 **G2**(结合性，associativity). 如果 π_1，π_2，$\pi_3 \in X$，那么

$$\pi_1 \circ (\pi_2 \circ \pi_3) = (\pi_1 \circ \pi_2) \circ \pi_3.$$

条件 **G3**(单位元，identity). 存在一个称为单位元的元素 $I \in X$，使得对于每一个 $\pi \in X$，有

$$I \circ \pi = \pi \circ I = \pi.$$

条件 **G4**(逆，inverse). 对于每一个 $\pi \in X$，存在一个称为 π 的逆的排列 $\pi^{-1} \in X$，使得

$$\pi \circ \pi^{-1} = \pi^{-1} \circ \pi = I.$$

为了验证这些条件，注意通过取 I 为下面的排列，**G3** 成立：

$$\begin{pmatrix} 1 & 2 & \cdots & n \\ 1 & 2 & \cdots & n \end{pmatrix}.$$

同样，如果我们取 π^{-1} 为逆转 π 的功能的排列那么 **G4** 成立，例如，如果

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 1 \end{pmatrix}$$

那么

$$\pi^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 3 & 2 \end{pmatrix}.$$

不言而喻，上面已经假设了 **G1**，它的验证及 **G2** 的验证是显然的.

450

如果 X 是任意集合，而 $\circ$ 定义 X 元素的积[⊖]，那么如果四个性质 $G1$，$G2$，$G3$ 和 $G4$ 成立，我们称序对 $G=(X, \circ)$为群(group). 我们给出若干群的例子. 如果 X 是正实数，且$a \circ b$意味着$a \times b$，那么对$(X, \circ)$是一个群. 公理 **G1** 成立，因为如果 a 和 b 是正实数，那么 $a \times b$ 总是正实数. 公理 **G2** 成立，因为有 $a \times (b \times c) = (a \times b) \times c$ 成立. 公理 **G3** 成立，因为我们可以取 I 为 1. 因为我们可以取 a^{-1}为 $1/a$，所以公理 **G4** 成立.

群的另外一个例子是$(X, \circ)$，其中 X 是全体实数，而 $a \circ b$ 定义为$a+b$. 公理 **G3** 的单位元是实数 0，而元素 a 的逆是$-a$. 注意当$a \circ b$ 定义为$a \times b$时，实数不是一个群. 因为唯一可能的单位元是 1. 但是数 0 没有逆. 不存在使得 $0^{-1} \times 0 = 1$ 的数 0^{-1}.

我们感兴趣的是排列的群，或称为排列群(permutation groups). 我们已观察到集合

⊖ 从技术上看，一个积 $\circ$ 是这样一个函数：它给 X 上的每一对元素a 和b 指定 X 中的另外一个元素$a \circ b$. (注意，我们或者定义 $a \circ b$ 总是 X 中的元素，或者显式地使条件 **G1** 成立.)

$A=\{1, 2, \cdots, n\}$的所有排列的全体定义一个群．这一排列群称为**对称群**（symmetric group）．排列群的另一个例子是下面的集合$\{1, 2, 3\}$的三个排列组成的：

$$\pi_1 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}, \quad \pi_2 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}, \quad \pi_3 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}. \tag{8.1}$$

留给读者证明它满足群的公理．

通常，物理对象或构形的对称性定义群，因此群理论在现代物理学中非常重要．作为一个例子，考虑在例 8.1 中所研究的 2×2 阵列的对称性，即旋转 0°，90°，180°和 270°．如果我们取 $a\circ b$ 为：首先进行对称转换 b，然后进行对称转换 a，那么这些对称性定义了一个群．例如，如果 a 是旋转 90°，b 是旋转 180°，那么 $a\circ b$ 是旋转 270°．

这个对称群可以认为是一个排列群，每一个对称转换都是$\{1, 2, 3, 4\}$的一个排列．为了明白其原因，如图 8.13 的第一部分所示的那样，标记 2×2 阵列中的四个格子．那么图 8.13 给出不同的对称转换所产生的结果标签．我们可以把这一标签考虑成对应于把标签 i 转换成 j 的排列．例如，我们可以把 90°旋转看成是把 1 变成 4，2 变成 1，3 变成 2 和 4 变成 3 的排列，即排列

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 3 \end{pmatrix}.$$

对应于其他旋转的排列如图 8.13 所示．

0°旋转		90°旋转		180°旋转		270°旋转	
2	1	1	4	4	3	3	2
3	4	2	3	1	2	4	1

$\pi_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}$　$\pi_2 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 3 \end{pmatrix}$　$\pi_3 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \end{pmatrix}$　$\pi_4 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}$

图 8.13　对应于 2×2 阵列旋转的排列

假设 A 是任意有限集合，而 f 是从 A 到 A 上的任意一一对应函数，于是正如我们前面所观察的那样，可以把 f 看成 A 的一个排列．如果 X 是这样的函数的集合，而$\circ$是函数的合成，那么 $G=(X, \circ)$是一个群，我们可以把 G 看成是一个排列群．例如，假设 $A=\{a, b, c\}$且 f，g 和 h 的定义如下：

451

$$f(a)=a, \quad f(b)=b, \quad f(c)=c;$$
$$g(a)=b, \quad g(b)=c, \quad g(c)=a;$$
$$h(a)=c, \quad h(b)=a, \quad h(c)=b.$$

于是 f，g 和 h 是一一对应函数．容易证明，如果 $X=\{f, g, h\}$，那么$(X, \circ)$是一个群．它是一个排列群．事实上，如果我们取 $a=1$，$b=2$，$c=3$，那么 f，g 和 h 分别是(8.1)式中的排列 π_1，π_2 和 π_3，所以这正是我们早前遇到的使用了不同表记法的排列群．

8.2.2 排列群衍生的等价关系

假设 $G=(X, \circ)$是集合 A 上的一个排列群．我们有时候使用 $\pi\in G$ 来表示 $\pi\in X$．我们可以通过下面的陈述定义 A 上的相同关系 S：

$$aSb \text{ 当且仅当存在一个 } G \text{ 中的排列使得 } \pi(a)=b \tag{8.2}$$

即 π 把 a 转变成 b．例如，如果 $A=\{1, 2, 3\}$且 G 是由(8.1)式的三个变换组成的，那么 $1S2$，因为 $\pi_2(1)=2$，而且 $3S2$，因为 $\pi_3(3)=2$．容易看到，对于该 S 和所有 a 和 b，aSb 成立．

定理 8.2　*如果 G 是集合 A 上的排列群，那么如(8.2)式所定义的 S 定义集合 A 上的等价关系．*

证明　我们需要证明 S 满足自反性、对称性和传递性. 因为恒等排列 I 在 G 中，对于所有 $a\in A$，有 $I(a)=a$，所以，对所有的 a，aSa 成立. 因此，自反性成立. 如果 aSb，那么 G 中存在 π 使得 $\pi(a)=b$. 现在，π^{-1} 在 G 中且 $\pi^{-1}(b)=a$. 我们得 bSa. 因此，对称性成立. 最后，假设 aSb 和 bSc 成立，那么在 G 中存在 π_1 和 π_2 使得 $\pi_1(b)=c$ 且 $\pi_2(a)=b$. 那么有 $\pi_1\circ\pi_2(a)=c$，所以 aSc 成立. ■

这一关系 S 称为*排列群 G 衍生的等价关系*(equivalence relation induced by the permutation group G).

[452]

下面给出更多的例子. 如果 G 是图 8.13 所示的集合{1, 2, 3, 4}的排列群，即是 2×2 阵列的旋转的群，那么对于{1, 2, 3, 4}中的所有 a 和 b，aSb 成立. 因此，S 有一个等价类{1, 2, 3, 4}. 接下来，假设 $A=\{1,2,3\}$ 且 G 是由下面的排列组成的：

$$\pi_1=\begin{pmatrix}1&2&3\\1&2&3\end{pmatrix}\quad 和 \quad \pi_2=\begin{pmatrix}1&2&3\\3&2&1\end{pmatrix}.$$

那么，G 是一个群(练习 3). 另外，S 下的等价类是{1, 3}和{2}. 回想一下，包含 a 的等价类 $C(a)$，是由 A 中所有满足 aSb 的 b 组成的，等价地，它是由 A 中所有使得对于 G 中的某个 π，有 $\pi(a)=b$ 成立的 b 组成的. 因此

$$C(a)=\{\pi(a):\pi\in G\}.$$

对于排列群这一特殊情况，$C(a)$ 有时候称为 a 的*轨迹*(orbit). 在我们刚给出的例子中，

$$C(1)=\{\pi_1(1),\pi_2(1)\}=\{1,3\}$$

是 1 的轨迹.

在计数不同构形的数量时，我们对相同关系下的(不同)等价类的数量的计数感兴趣. 计数的方法之一是简单地计算所有等价类并枚举出它们. 然而这通常不切实际. 下一节给出不用枚举计数等价类数量的方法.

8.2.3 图的自同构

设 H 是一个固定的无标签图[⊖]. H 的*自同构*(automorphism)是 H 的顶点的满足下面条件的排列 π：如果 $\{x,y\}\in E(H)$，那么 $\{\pi(x),\pi(y)\}\in E(H)$. 使用 3.1.3 节的术语，自同构是图到其自身的同构. 例如，考虑图 8.14 中的图. 我们可以如图所示的那样，标记这些顶点为 1, 2, 3, 4, 5, 并取 $\pi(1)=4$, $\pi(2)=5$, $\pi(3)=1$, $\pi(4)=2$, $\pi(5)=3$ 来定义一个自同构. 这与 144°的旋转相同. 第二个自同构可以通过取 $\pi=\begin{pmatrix}1&2&3&4&5\\1&5&4&3&2\end{pmatrix}$ 而得到. 这相当于以连接顶点 1 与边{3, 4}中点的直线为轴的反射.

图 8.14　由 $\pi(1)=4$, $\pi(2)=5$, $\pi(3)=1$, $\pi(4)=2$, $\pi(5)=3$ 给出的自同构

作为第二个例子，考虑图 8.15 所示的图 $K_{1,3}$. 如图所示，使用整数 1, 2, 3, 4 标记顶点. $K_{1,3}$ 的自同构的例子之一是通过顺时针旋转该图 120°得到的. 这对应于排列 $\begin{pmatrix}1&2&3&4\\1&3&4&2\end{pmatrix}$. 通过边{1, 2}的反射产生自同构 $\begin{pmatrix}1&2&3&4\\1&2&4&3\end{pmatrix}$. 一共存在 6 个自同构：恒等自同构，通过 120°和 240°旋转得到的自同构，通过三条边的反射而得到的(三个)自同构. 我们是如何知道不存在其他自同构的呢？显然，每一个自同构都把 1 变成 1. 因此，我们需要寻找{1, 2, 3, 4}的所有把 1

[453]

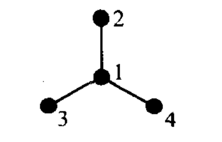


图 8.15　图 $K_{1,3}$

⊖ 在本章，G 总表示一个群而 H 总表示一个图.

变成1的排列，而这些排列有3!=6个.（8.2)式的相同关系给出两个等价类{1}和{2，3，4}.

定理8.3 一个图的所有自同构的集合是排列群.

证明 我们把它留给读者(练习26). ■

我们使用Aut(H)来表示图H的自同构群. 因此，例如，Aut(K_n)是{1，2，…，n}上的对称群，因为K_n的顶点的所有排列定义了自同构. 更多关于图的自同构群的内容可以参见Cameron[1983]、Gross和Yellen[1999].

本节练习

1. 以下面所示的形式写出所给的每一个排列：

$$\begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ a_1 & a_2 & a_3 & \cdots & a_n \end{pmatrix}.$$

(a)7263541 (b)52413 (c)67583412

2. 对于如下所示的排列π_1和π_2，给出$\pi_1 \circ \pi_2$.

(a)$\pi_1=\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix}$，$\pi_2=\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 1 & 4 \end{pmatrix}$

(b)$\pi_1=\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \end{pmatrix}$，$\pi_2=\begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 1 & 2 \end{pmatrix}$

454

(c)$\pi_1=\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 5 & 4 \end{pmatrix}$，$\pi_2=\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 5 & 1 & 3 \end{pmatrix}$

(d)$\pi_1=\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 1 & 4 & 5 & 6 & 2 \end{pmatrix}$，$\pi_2=\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 1 & 3 & 6 & 5 \end{pmatrix}$

3. 假设$A=\{1, 2, 3\}$而X是下面排列的集合：

$$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \text{ 和 } \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}.$$

如果∘是合成，那么证明(X，∘)是群.

4. 假设$A=\{1, 2, 3\}$，而X是在等式(8.1)中给出的三个排列组成的集合. 证明在合成下X定义一个群.

5. 对于下列的X和∘，检验哪些满足群的四个公理.

(a)X=排列$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix}$和$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 3 & 2 & 1 \end{pmatrix}$，而∘=合成.

(b)X=排列$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix}$和$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 3 & 2 & 4 & 5 \end{pmatrix}$，而∘=合成.

(c)$X=\{0, 1\}$而∘是由下面的规则定义的：$0\circ0=0$，$0\circ1=1$，$1\circ0=1$，$1\circ1=1$.

(d)X=有理数，∘=加法.

(e)X=有理数，∘=乘法.

(f)X=负实数，∘=加法.

(g)X=所有实数2×2矩阵，∘=矩阵乘法.

6. 证明函数$\{f, g\}$的集合是集合$A=\{x, y, u, v\}$上的排列的群，如果$f(x)=v$，$f(y)=u$，$f(u)=y$，$f(v)=x$，$g(x)=x$，$g(y)=y$，$g(u)=u$，$g(v)=v$.

7. 如果用$f(x)=y$，$f(y)=x$，$f(u)=v$，$f(v)=u$重新定义f，那么练习6的结论仍然为真吗？

8. 假设$A=\{1, 2, 3, 4, 5, 6\}$且G是下面排列的群：

$$\left\{\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 7 & 6 & 5 & 4 & 3 & 2 & 1 \end{pmatrix}\right\}.$$

如果 S 是由 G 衍生的等价关系，那么：

(a)$1S2$ 成立吗？ (b)$3S5$ 成立吗？ (c)$5S6$ 成立吗？

9. 对于练习 8，确定轨迹 $C(1)$ 和 $C(4)$.

10. 对于如下所示的 A 和 G，确定由 G 衍生的等价关系 S 下的等价类：

(a)$A=\{1, 2, 3, 4, 5\}$,

$$G=\left\{\begin{pmatrix}1&2&3&4&5\\1&2&3&4&5\end{pmatrix}, \begin{pmatrix}1&2&3&4&5\\5&4&3&2&1\end{pmatrix}\right\}$$

455

(b)$A=\{1, 2, 3, 4, 5, 6\}$,

$$G=\left\{\begin{pmatrix}1&2&3&4&5&6\\1&2&3&4&5&6\end{pmatrix}, \begin{pmatrix}1&2&3&4&5&6\\2&1&3&4&5&6\end{pmatrix}, \begin{pmatrix}1&2&3&4&5&6\\1&2&4&3&5&6\end{pmatrix},\right.$$

$$\begin{pmatrix}1&2&3&4&5&6\\1&2&3&4&6&5\end{pmatrix}, \begin{pmatrix}1&2&3&4&5&6\\2&1&4&3&5&6\end{pmatrix}, \begin{pmatrix}1&2&3&4&5&6\\2&1&3&4&6&5\end{pmatrix},$$

$$\left.\begin{pmatrix}1&2&3&4&5&6\\1&2&4&3&6&5\end{pmatrix}, \begin{pmatrix}1&2&3&4&5&6\\2&1&4&3&6&5\end{pmatrix}\right\}$$

(c)$A=\{1, 2, 3, 4, 5\}$,

$$G=\left\{\begin{pmatrix}1&2&3&4&5\\1&2&3&4&5\end{pmatrix}, \begin{pmatrix}1&2&3&4&5\\1&3&2&4&5\end{pmatrix}, \begin{pmatrix}1&2&3&4&5\\5&2&3&4&1\end{pmatrix}, \begin{pmatrix}1&2&3&4&5\\5&3&2&4&1\end{pmatrix}\right\}$$

11. 考虑 8.1 节练习 9 所描述的 2×2 阵列的所有对称的集合. 如果 π_1 和 π_2 是下列对称，确定 $\pi_1\circ\pi_2$.

(a)π_1＝旋转 90°，π_2＝相对于水平线的反射.

(b)π_1＝相对于垂线的反射，π_2＝旋转 180°.

(c)π_1＝旋转 270°，π_2＝相对于垂线的反射.

(d)π_1＝旋转 180°，π_2＝相对于从左下角到右上角的对角线的反射.

12. 继续练习 11，把下列的对称描述为集合$\{1, 2, 3, 4\}$的排列：

(a)相对于水平线的反射.

(b)相对于垂线的反射.

(c)相对于从左下角到右上角的对角线的反射.

(d)相对于从左上角到右下角的对角线的反射.

13. 继续练习 11，所有的对称(旋转和反射)的集合是群吗？

14. 对于例 8.1，假设我们对任意格子可以使用 c 种颜色中的任意一种颜色.

(a)仅允许旋转时，有多少种不同的着色？

(b)如果允许旋转及相对于垂直、水平和对角线的反射，有多少种不同的着色？

15. 寻找正文中没有提及的对应于图 $K_{1,3}$ 的每一个自同构的排列.

16. 对于 8.1 节练习 17，如果 π_1 和 π_2 分别旋转 180°和 270°，那么确定 $\pi_1\circ\pi_2$.

17. 在 8.1 节练习 19 的情况下，对于下列各 π_1 和 π_2，确定 $\pi_1\circ\pi_2$.

(a)π_1 和 π_2 分别是 180°的旋转和相对于对边中点的水平连线的反射.

(b)π_1 和 π_2 分别是相对于对边中点的垂直连线的反射和 90°的旋转.

(c)π_1 和 π_2 分别是相对于连接对角的正斜对角线(/)的反射和相对于对边中点的水平连线的反射.

456

18. 在 8.1 节练习 19 的情况下，所有对称的集合是群吗？

19. 如果 π_1 和 π_2 是排列，$\pi_1\circ\pi_2$ 不一定等于 $\pi_2\circ\pi_1$(因此，我们说排列的积并不一定是可交换的).

(a)使用 $\pi_1=\begin{pmatrix}1&2&3&4\\1&4&2&3\end{pmatrix}$和 $\pi_2=\begin{pmatrix}1&2&3&4\\2&1&3&4\end{pmatrix}$进行说明.

(b)确定 2×2 阵列的两个对称 π_1 和 π_2(π_1 和 π_2 可以是旋转也可以是反射)使得 $\pi_1\circ\pi_2\neq\pi_2\circ\pi_1$.

20. 证明：对于所有素数 p，整数集合$\{1, 2, \cdots, p-1\}$以及模 p 的乘法 $\circ$ 形成一个群.

21. 在练习 20 中，如果 p 不是素数，我们还能得到一个群吗？为什么？

22. 假设 G 是一个排列群. 固定 G 中的一个排列 σ. 如果 π_1 和 π_2 都在 G 中，我们说 $\pi_1 S \pi_2$，如果 $\pi_1 = \sigma^{-1} \circ (\pi_2 \circ \sigma)$. 证明 S 是一个等价关系.
23. (a)确定 $\mathrm{Aut}(L_4)$，其中 L_4 是四顶点的链.
    (b)确定 $\mathrm{Aut}(Z_4)$，其中 Z_4 是四顶点的回路.
    (c)确定 $\mathrm{Aut}(K_4-K_2)$，其中 K_4-K_2 是图 8.16 所示的图.

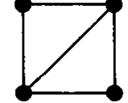


图 8.16 K_4-K_2

24. 描述 $\mathrm{Aut}(Z_n)$并确定 Z_n 的自同构的数量.
25. 图 $K_{m,n}$在第一个类中有 m 个顶点，在第二个类中有 n 个顶点，且在不同类的所有顶点对之间有边. 图 8.15的 $K_{1,3}$是一个特殊情况.
    (a)如果 $m \neq n$，描述 $\mathrm{Aut}(K_{m,n})$并确定 $K_{m,n}$的自同构的数量.
    (b)对于 $m=n$，重复(a).
26. 证明定理 8.3.

8.3 伯恩赛德引理

8.3.1 伯恩赛德引理的陈述

本节给出排列群衍生的等价关系下的(不同)等价类的计数方法. 假设 G 是集合 A 的一个排列群. A 中的元素 a 在 G 的排列 π 下称为不变的(invariant)(或不动的(fixed))，如果 $\pi(a)=a$. 设 $\mathrm{Inv}(\pi)$是 A 在 π 之下不变元素的数量.

457

定理 8.4(伯恩赛德引理)[⊖] *设 G 是集合 A 的一个排列群，而且设 S 是 G 衍生的 A 上的等价关系. 那么 S 中等价类的数量由下式给出：*

$$\frac{1}{|G|}\sum_{\pi \in G}\mathrm{Inv}(\pi).$$

为了说明这一定理，我们首先考虑集合 $A=\{1, 2, 3\}$和由等式(8.1)所定义的 A 的排列群. 这时，$\mathrm{Inv}(\pi_1)=3$. 因为 1、2 和 3 在 π_1 下是不变的，而且 $\mathrm{Inv}(\pi_2)=\mathrm{Inv}(\pi_3)=0$，因为在 π_2 或 π_3 下没有不变元素. 因此，衍生等价关系 S 下的等价类的数量由$\frac{1}{3}(3+0+0)=1$ 给出. 这是正确的，因为对于所有的 a，$b\in A$，aSb 成立. 只存在一个等价类$\{1, 2, 3\}$.

作为第二个例子，假设 $A=\{1, 2, 3, 4\}$且 G 是由下列排列组成的：

$$\pi_1=\begin{pmatrix}1&2&3&4\\1&2&3&4\end{pmatrix},\quad \pi_2=\begin{pmatrix}1&2&3&4\\2&1&3&4\end{pmatrix},$$
$$\pi_3=\begin{pmatrix}1&2&3&4\\1&2&4&3\end{pmatrix},\quad \pi_4=\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}. \tag{8.3}$$

容易验证 G 是群. 此时，$\mathrm{Inv}(\pi_1)=4$，$\mathrm{Inv}(\pi_2)=2$，$\mathrm{Inv}(\pi_3)=2$，$\mathrm{Inv}(\pi_4)=0$，而且衍生等价关系 S 下的等价类数量是$\frac{1}{4}(4+2+2+0)=2$. 这是正确的，因为两个等价类是$\{1, 2\}$和$\{3, 4\}$(练习 1).

作为第三个例子，考虑$\{1, 2, 3\}$上所有弱序的集合，如图 8.9 所示. $\{1, 2, 3\}$的每一个排列 π，通过用 $\pi(i)$取代弱序中的每一个元素 i，衍生 A 的排列 π^*. 例如，如果 $\pi=\begin{pmatrix}1&2&3\\2&3&1\end{pmatrix}$，那么对于弱序 J，$\pi^*(J)=K$，而 J 和 K 如图 8.9 所示. 所有 π^* 的集合定义 A 的一个排列群G. 在 G 中存在 6 个排列，下面 π_1 到 π_6 中的每一个对应一个排列：$\pi_1=\begin{pmatrix}1&2&3\\1&2&3\end{pmatrix}$、$\pi_2=$

⊖ 本引理的这一版本是 Burnside[1911]所给出的关键引理的一个简单推论，而且通常称此版本为伯恩赛德(Burnside)引理.

$\begin{pmatrix}1 & 2 & 3\\ 1 & 3 & 2\end{pmatrix}$、$\pi_3=\begin{pmatrix}1 & 2 & 3\\ 2 & 1 & 3\end{pmatrix}$、$\pi_4=\begin{pmatrix}1 & 2 & 3\\ 2 & 3 & 1\end{pmatrix}$、$\pi_5=\begin{pmatrix}1 & 2 & 3\\ 3 & 1 & 2\end{pmatrix}$、$\pi_6=\begin{pmatrix}1 & 2 & 3\\ 3 & 2 & 1\end{pmatrix}$. 如果我们使用例 8.6 中定义的相同关系，那么 $\text{Inv}(\pi_1^*)=13$，因为 π_1^* 使图 8.9 中的所有 13 个弱序不变. $\text{Inv}\pi_2^*=3$，因为 π_2^* 使弱序 I，J 和 M 不变. 类似地，$\text{Inv}(\pi_3^*)=\text{Inv}(\pi_6^*)=1$. 最后，$\text{Inv}\pi_4^*=\text{Inv}\pi_5^*=1$，因为 π_4^* 和 π_5^* 仅使 M 不变. 根据伯恩赛德引理，弱序的等价类的数量由下式给出：

$$\frac{1}{6}[13+3+3+1+1+3]=4.$$

458

这些等价类分别是$\{A, B, C, D, E, F\}$，$\{G, H, I\}$，$\{J, K, L\}$和$\{M\}$.

作为第四个例子，考虑自同构群 $G=\text{Aut}(K_{1,3})$. 根据我们在 8.2.3 节的讨论，$|G|=6$. 对于恒等变换 π，$\text{Inv}(\pi)=4$，对于旋转 120°和 240°的变换，$\text{Inv}(\pi)=1$，对于相对于一条边的三个反射变换，$\text{Inv}(\pi)=2$. 因此有

$$\frac{1}{|G|}\sum_{\pi\in G}\text{Inv}(\pi)=\frac{1}{6}[4+1+1+2+2+2]=2,$$

这一计算结果与$\{1\}$和$\{2, 3, 4\}$是 S 下的等价类的结论一致.

在 8.4 节中，我们将看到如何把伯恩赛德引理运用到诸如例 8.1 至例 8.5 这样的例子中.

8.3.2 伯恩赛德引理的证明[⊖]

现在我们给出伯恩赛德引理的证明. 假设 G 是集合 A 上的一个排列群. 对于每一个 $a\in A$，设 a 的稳定算子(stabilizer)$\text{St}(a)$是 G 中使 a 不变的所有排列的集合，即 $\text{St}(a)=\{\pi\in G: \pi(a)=a\}$. 设 $C(a)$是 a 的轨迹，即衍生等价关系 S 下包含 a 的等价类，即满足下面条件的所有元素 b 的集合：对于 G 中的某个 π 有 $\pi(a)=b$. 作为说明，假设 $A=\{1, 2, 3\}$，且 G 是由等式(8.1)所定义的，那么 $C(2)=\{\pi_1(2), \pi_2(2), \pi_3(2)\}=\{1, 2, 3\}$. 而且有 $\text{St}(2)=\{\pi_1\}$.

引理 8.1　*假设 G 是集合 A 上的一个排列群，而且 a 在 A 中. 那么*

$$|\text{St}(a)|\cdot|C(a)|=|G|.$$

证明　假设 $C(a)=\{b_1, b_2, \cdots, b_r\}$. 那么存在把 a 传递到 b_1 的排列 π_1(也可能存在其他把 a 传递到 b_1 的排列，但是我们从中挑选出一个这样的排列). 同样也存在把 a 传递到 b_2 的排列 π_2，把 a 传递到 b_3 的排列 π_3，以此类推. 设 $P=\{\pi_1, \pi_2, \cdots, \pi_r\}$. 注意 $|P|=|C(a)|$. 我们要证明对于 G 中的每一个排列 π 正好能用一种方法写成 P 中排列和 $\text{St}(a)$中排列的积. 那么，根据乘法规则有 $|G|=|P|\cdot|\text{St}(a)|=|C(a)|\cdot|\text{St}(a)|$.

给定 G 中的 π，注意，对某个 k 有 $\pi(a)=b_k$. 因此，$\pi(a)=\pi_k(a)$，所以有 $\pi_k^{-1}\circ\pi$ 使得 a 不变. 因此 $\pi_k^{-1}\circ\pi$ 在 $\text{St}(a)$中. 但是，我们有

$$\pi_k\circ(\pi_k^{-1}\circ\pi)=(\pi_k\circ\pi_k^{-1})\circ\pi=I\circ\pi=\pi,$$

所以，π 是 P 中排列和 $\text{St}(a)$中排列的积.

其次，假设 π 可以用两种方法写成 P 中排列和 $\text{St}(a)$中排列的积；即假设 $\pi=\pi_k\circ\gamma=\pi_l\circ\delta$，其中 γ 和 δ 在 $\text{St}(a)$中. 现在有$(\pi_k\circ\gamma)(a)=b_k$ 和$(\pi_l\circ\delta)(a)=b_l$. 因为 $\pi_k\circ\gamma=\pi_l\circ\delta$，$b_k$ 一定等于 b_l，所以 $k=l$. 因此，在等式 $\pi_k\circ\gamma=\pi_k\circ\delta$ 的两边乘以 π_k^{-1}，我们得到 $\gamma=\delta$. ■

459

为了说明这一引理，设 $A=\{1, 2, 3\}$，且设 G 是由(8.1)式所定义的. 根据我们上面的计算，有 $C(2)=\{1, 2, 3\}$而 $\text{St}(2)=\{\pi_1\}$. 因此，有

$$|G|=3=(1)\cdot(3)=|\text{St}(2)|\cdot|C(2)|.$$

为了完成伯恩赛德引理的证明，我们证明：如果 $A=\{1, 2, \cdots, n\}$且 $G=\{\pi_1, \pi_2, \cdots,$

⊖ 本小节可以跳过.

$\pi_m\}$，那么有

$$\mathrm{Inv}(\pi_1)+\mathrm{Inv}(\pi_2)+\cdots+\mathrm{Inv}(\pi_m)=|\mathrm{St}(1)|+|\mathrm{St}(2)|+\cdots+|\mathrm{St}(n)|.$$

这是成立的，因为这一等式的两边都计数使得 $\pi(a)=a$ 的序对 (a,π) 的数量. 根据引理 8.1，有

$$\frac{1}{|G|}[\mathrm{Inv}(\pi_1)+\mathrm{Inv}(\pi_2)+\cdots+\mathrm{Inv}(\pi_m)]=\frac{1}{|C(1)|}+\frac{1}{|C(2)|}+\cdots+\frac{1}{|C(n)|}. \quad (8.4)$$

注意，x 总是在 $C(x)$ 中，因为 $I(x)=x$. 因此，根据定理 8.1，$C(x)=C(y)$ 当且仅当 x 在 $C(y)$ 中. 因此，如果 $C(x)=\{b_1, b_2, \cdots, b_k\}$，那么正好存在 k 个等于 $C(x)$ 的等价类 $C(b_1)$，$C(b_2)$，…，$C(b_k)$. 我们可以把这些等价类分割成诸如 $\{C(b_1), C(b_2), \cdots, C(b_k)\}$ 的群，每一个群是一组相等的等价类. 注意 $|C(b_i)|=k$. 因此，

$$\frac{1}{|C(b_1)|}+\frac{1}{|C(b_2)|}+\cdots+\frac{1}{|C(b_k)|}=\frac{1}{k}+\frac{1}{k}+\cdots+\frac{1}{k}=1.$$

因此(8.4)式的右边的和计数不同等价类的数量，所以这个数也由(8.4)式的左边给出. 因此，伯恩赛德引理成立.

为了说明这一证明，假设 $A=\{1, 2, 3, 4\}$，而 G 由(8.3)式的四个排列给出. 那么 $C(1)=\{1, 2\}$，$C(2)=\{1, 2\}$，$C(3)=\{3, 4\}$，$C(4)=\{3, 4\}$. 因此，

$$\frac{1}{|C(1)|}+\frac{1}{|C(2)|}=1,$$

$$\frac{1}{|C(3)|}+\frac{1}{|C(4)|}=1$$

和

$$\frac{1}{|C(1)|}+\frac{1}{|C(2)|}+\frac{1}{|C(3)|}+\frac{1}{|C(4)|}=2,$$

这是等价类的数量.

本节练习

1. 验证(8.3)式的四个排列定义一个群，并且验证它衍生的等价关系下的等价类是 $\{1, 2\}$ 和 $\{3, 4\}$.

460

2. 对于 $\mathrm{Aut}(K_{1,3})$ 中的每一个自同构 a，计算 $\mathrm{St}(a)$，$C(a)$ 并验证引理 8.1.
3. 通过计算下式，对 $\mathrm{Aut}(K_{1,3})$ 验证伯恩赛德引理的证明：
$$\frac{1}{|C(1)|}+\frac{1}{|C(2)|}+\frac{1}{|C(3)|}+\frac{1}{|C(4)|}.$$
4. 对于 8.2 节练习 8，使用伯恩赛德引理确定 S 下的等价类的数量.
5. 对于 8.2 节练习 10 的每一种情况，使用伯恩赛德引理确定 S 下的等价类的数量并通过计算等价类加以验证.
6. 对于 8.2 节练习 10 的每一种情况，设 $a=1$.
   (a)确定 $\mathrm{St}(a)$.　　(b)确定 $C(a)$.　　(c)验证引理 8.1.
7. 对于 8.2 节练习 10 的每一种情况，验证
$$\frac{1}{|C(1)|}+\frac{1}{|C(2)|}+\cdots$$
   给出 S 下的等价类的数量.
8. 对于图 8.14 中的图的每一个自同构 a，计算 $\mathrm{St}(a)$ 和 $C(a)$ 并验证引理 8.1.
9. 对于图 8.14 中的图 H，通过计算下式并比较 S 下的等价类数量，对 $\mathrm{Aut}(H)$ 验证伯恩赛德引理的证明：
$$\frac{1}{|C(1)|}+\frac{1}{|C(2)|}+\frac{1}{|C(3)|}+\frac{1}{|C(4)|}+\frac{1}{|C(5)|}.$$
10. 当如例 8.6 那样定义相同时，使用伯恩赛德引理计算 $\{1, 2, 3, 4\}$ 上的弱序的等价类数量.
11. 假设在一个圆桌的周围要安排 5 名谈判代表的固定席位. 如果沿着圆桌的座位旋转不认为是改变坐席

安排，那么利用伯恩赛德引理计算 5 名谈判代表的坐席安排的不同方法的数量.

12. 假设：

$$A=\{1,2,3,4\},\quad G=\left\{\begin{pmatrix}1&2&3&4\\1&2&3&4\end{pmatrix},\begin{pmatrix}1&2&3&4\\2&1&3&4\end{pmatrix},\begin{pmatrix}1&2&3&4\\1&2&4&3\end{pmatrix}\right\}.$$

$|St(1)|\cdot|C(1)|=|G|$ 成立吗？解释发生了什么？

13. 假设我们使用标签 1，2，…，n 标记一个图 H 的 n 个顶点. H 的任意标签可以认为是$\{1, 2, \cdots, n\}$的一个排列，如果我们开始于一个固定的标签. 例如，如果我们开始于如图 8.14 所示的原始标签，那么图 8.17 所示的新标签对应于下面的排列：

$$\begin{pmatrix}1&2&3&4&5\\2&1&5&3&4\end{pmatrix}.$$

图 8.17　图 8.14 中的图的一个新标签

H 的每一个自同构 π 可以认为是把 H 的任意一个标签 σ 转换成另一个标签：我们只使用标签 $\pi\circ\sigma$. 因此，H 的不同标签的数量对应于$\{1, 2, \cdots, n\}$排列集合上自同构群 $\mathrm{Aut}(H)$衍生的等价关系下的等价类的数量.

461

(a)证明不同标签的数量由 $n!/|\mathrm{Aut}(H)|$ 给出.

(b)如果 L_4 是四顶点的链，根据(a)的结果确定不同标签的数量并通过枚举这些标签加以验证.

(c)对于 Z_4，长度为 4 的回路重复(b).

(d)对于图 8.15 中的图 $K_{1,3}$，重复(b).

14. 利用本节的方法，完成第 2 章的附加练习 25.

8.4 不同的着色

8.4.1 着色的定义

假设 D 是对象的集合，D 的一个着色(coloring)是给 D 中的每一个对象分配一种颜色. 在这一意义下，如果 D 是一个图的顶点集合，那么一个着色就是给每一个顶点分配一种颜色，与第 3 章中所使用的规则无关，即如果 x 和 y 被一条边连接着，那么它们必须得到不同的颜色. 一个着色可以认为是一个函数 $f: D\rightarrow R$，其中 R 是颜色集合. 如果 D 有 n 个元素，且 R 有 m 个元素，那么存在 m^n 个 D 的着色.

在例 8.1 中，集合 D 是 2×2 阵列中的四个格子的集合，而集合 R 是集合{黑色，白色}. 在例 8.2 中，集合 D 可以认为是整数 1，2，…，k，代表珠子的 k 个空间，而 R 是集合$\{b, r\}$. 在例 8.3 中，D 是长度为 n 的位串集合，而 $R=\{0, 1\}$. 在例 8.4 中，D 是 7 顶点树的顶点集合，而 R={黑色，白色}. 最后，在例 8.5 中，集合 D 由正四面体的顶点 a，b，c，d 组成，而集合 R 是集合$\{CH_3, C_2H_5, H, Cl\}$.

顶点集合 $V=\{1, 2, \cdots, p\}$上的每一个图都可以认为是一个着色. 取 D 是 V 的所有 2 元素子集的集合，R 是$\{0, 1\}$，并设 $f(\{i, j\})$是 1，如果$\{i, j\}\in E(G)$，否则是 0. 练习 21 和练习 22 利用这一概念计算 p 个顶点的不同(不同构的)图的数量.

在我们所有的例子中，我们也允许 D 的元素的某些排列. 这些排列定义一个群 G. 特别地，在例 8.1 中，我们允许有图 8.13 给出的四个旋转，它们定义一个排列群 G. 在例 8.2 中，在有两个珠子的情况下，所给的排列是

462

$$\pi_1=\begin{pmatrix}1&2\\1&2\end{pmatrix}\quad 和\quad \pi_2=\begin{pmatrix}1&2\\2&1\end{pmatrix}.$$

注意 π_1 和 π_2 定义一个群——群 G. 更一般地，如果有 k 个珠子，那么群 G 是由下面两个排列组成的群：

$$\begin{pmatrix}1&2&\cdots&k\\1&2&\cdots&k\end{pmatrix}\quad 和\quad \begin{pmatrix}1&2&\cdots&k\\k&k-1&\cdots&1\end{pmatrix}.$$

在例 8.5 中，群 G 中的排列对应于我们在 8.1 节所描述的正四面体的对称变换. 我们将在 8.5 节讨论这一例子，在那里，我们将给出表示这些排列的简单方法.

例 8.4 中的群是什么呢？假设我们从图 8.18 给出的第一棵标签树开始，即其下是 π_1 的标签树. 这时，这棵树的任意其他标签对应于{1，2，…，7}的一个排列. 并非{1，2，…，7}的每一个排列都对应于左、右孩子互换的意义下认为是等价的标签. 图 8.18 给出所有从第一棵树通过交换左、右孩子而得到的标签树. 例如，其下是 π_2 的标签树是通过交换顶点 4 和 5 而得到的. 其下是 π_3 的标签树是通过交换顶点 6 和 7 得到的. 其下是 π_4 的标签树是通过由顶点 2、4、5 所生成的子树 T_1 与由顶点 3、6、7 所生成的子树 T_2 进行交换而得到的. 其下是 π_5 的标签树是通过顶点 4、5 与顶点 6、7 交换得到的. 其下是 π_6 的标签树是通过首先交换子树 T_1 和 T_2，然后再交换顶点 6 和 7 而得到的，等等. 这 8 棵树对应于 D 的元素的合法排列，也就是 G 的成员. 对于每一棵标签树的排列如图 8.18 所示. 注意

$$\pi_4 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 3 & 2 & 6 & 7 & 4 & 5 \end{pmatrix},$$

因为顶点 1 保持不变，顶点 2 被变成顶点 3，顶点 3 被变成顶点 2 等等. G 是群的验证留给读者(练习 19). 不难证明，我们描述的这些排列正好是图 8.18 所示的第一棵树的自同构.

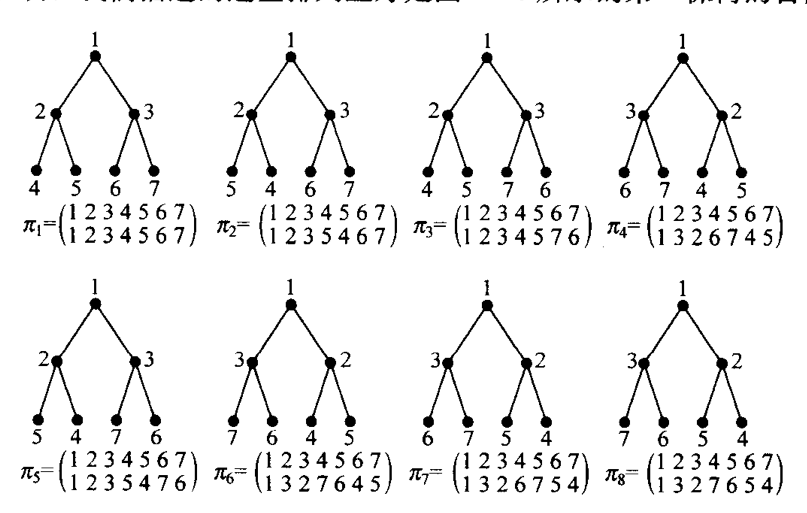


图 8.18　一棵标签树及通过交换左、右孩子所得到的 8 棵标签子树

我们将简短地讨论例 8.3 中的开关函数. 在例 8.1～例 8.5 中，我们所感兴趣的是确定两个着色是否是不同的着色，以及计数着色的等价类数量. 然而，两个相同着色的等价关系不同于排列群 G 衍生的等价关系 S. 因为 S 是集合 D 上的关系，而不是 D 的着色集合上的关系. 因此，直接使用伯恩赛德引理不能帮助我们计数着色等价类的数量. 在 8.4.2 节，我们讨论如果定义适当的等价关系.

8.4.2　等价着色

假设 $C(D, R)$ 是使用 R 中颜色的 D 的所有着色的集合，而 G 是集合 D 的一个排列群，π 在 G 中. 对应于 π 的是 $C(D, R)$ 中的一个排列 π^*. π^* 把 $C(D, R)$ 中的每一个着色变成另一个着色. 如果 f 是一个着色，那么新着色 $\pi^* f$ 定义为 $(\pi^* f)(a)$ 是 $f(\pi(a))$. 即 $\pi^* f$ 分配给 a 的颜色与 f 分配给 $\pi(a)$ 的颜色相同. 在例 8.1 中，π^* 把 2×2 阵列的一个给定的着色 C_i 转换成另一个着色. 例如，假设 π_2 是这个阵列的 90°旋转，如图 8.13 所示，我们来计算 π_2^*，如果 C_1，C_2，…如图 8.10 所示，首先注意到对于所有的 x，有 $C_1(x)$ = 黑色，所以 $(\pi_2^* C_1)(a)$ =

$C_1(\pi_2(a))$＝黑色．这意味着对于所有的 a，$(\pi_2^*)(a)$＝黑色，所以 $\pi_2^* C_1$ 是着色 C_1．接下来，因为 $4=\pi_2(1)$，$(\pi_2^* C_2)(1)=C_2(\pi_2(1))=C_2(4)$＝黑色．同样，$(\pi_2^* C_2)(2)=C_2(\pi_2(2))=C_2(1)$＝黑色，$(\pi_2^* C_2)(3)=C_2(\pi_2^*(3))=C_2(2)$＝白色，而且 $(\pi_2^* C_2)(4)=C_2(\pi_2(4))=C_2(3)$＝黑色．因此，$\pi_2^* C_2$ 与着色 C_3 是同样的．类似地，$\pi^* C_3=C_4$，$\pi_2^* C_4=C_5$ 等等．总之，排列 π_2^* 是由下式给出的：

$$\pi_2^* = \begin{pmatrix} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 & C_9 & C_{10} & C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_1 & C_3 & C_4 & C_5 & C_2 & C_7 & C_8 & C_9 & C_6 & C_{11} & C_{10} & C_{13} & C_{14} & C_{15} & C_{13} & C_{16} \end{pmatrix}. \quad (8.5)$$

类似地，如果 π_1 是 0°的旋转，π_3 是 180°的旋转，π_4 是 270°的旋转，那么

463 ~ 464

$$\pi_1^* = \begin{pmatrix} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 & C_9 & C_{10} & C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 & C_9 & C_{10} & C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \end{pmatrix} \quad (8.6)$$

$$\pi_3^* = \begin{pmatrix} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 & C_9 & C_{10} & C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_1 & C_4 & C_5 & C_2 & C_3 & C_8 & C_9 & C_6 & C_7 & C_{10} & C_{11} & C_{14} & C_{15} & C_{12} & C_{13} & C_{16} \end{pmatrix}, \quad (8.7)$$

以及

$$\pi_4^* = \begin{pmatrix} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 & C_9 & C_{10} & C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_1 & C_5 & C_2 & C_3 & C_4 & C_9 & C_6 & C_7 & C_8 & C_{11} & C_{10} & C_{15} & C_{12} & C_{13} & C_{14} & C_{16} \end{pmatrix}. \quad (8.8)$$

因此，D 的排列群 G 对应于 $C(D, R)$ 的排列群 G^*；$G^*=\{\pi^*: \pi\in G\}$（为什么 G^* 是一个群）．注意，G 和 G^* 的元素数量相同．另外，如果 S^* 是由 G^* 衍生的等价关系，那么 S^* 是我们所感兴趣的相同关系．在 S^* 下，两个着色 f 和 g 被认为是等价的，如果对于 D 中的某个排列 π，有 $g=\pi^* f$，即对于 D 中的所有 a，有 $g(a)=f(\pi(a))$．

在例子 8.2 中，假设

$$\pi = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}.$$

设 $f(1)=r$，$f(2)=b$，$g(1)=b$，$g(2)=r$．这时，对于所有的 a，有 $g(a)=f(\pi(a))$，所以有 $fS^* g$．这恰好说明两个着色 rb 和 br 是等价的．今后，为了区别 S 下与 S^* 下的等价类，我们称后者的等价类为*模式*(pattern)．我们对计算不同模式的数量感兴趣．这可以通过把伯恩赛德引理运用于 G^* 来完成．

在例 8.1 中，$\mathrm{Inv}(\pi_1^*)=16$，$\mathrm{Inv}(\pi_2^*)=2$，$\mathrm{Inv}(\pi_3^*)=4$，和 $\mathrm{Inv}(\pi_4^*)=2$，因为 π_1^* 使所有 16 种着色 C_i 不变，π_2^* 和 π_4^* 只使 C_1 和 C_{16} 不变，而 π_3^* 使 C_1，C_{10}，C_{11} 和 C_{16} 不变，因此，S^* 下的等价类数量是 $\frac{1}{4}(16+2+4+2)=6$，这一结果与图 8.10 一致[我们可以直接计算 $\mathrm{Inv}(\pi_i^*)$，而无需先计算 π_i^*．例如，π_3^* 只使下面的这些着色不变：在格子 1 和格子 3 中相同的着色以及在格子 2 和格子 4 中相同的着色．因为在这样的着色中，对于格子 1 和 3 存在 2 种着色选择，且对格子 2 和 4 存在 2 种着色选择，因此，对于这一着色有 $2^2=4$ 种颜色选择．类似地，π_2^* 仅使四个格子都有相同颜色的着色不变，因为每一个格子的颜色必须与顺时针方向旋转 90°后的颜色相同．因此，只存在 2 种着色]．

对于项链的情况(例 8.2)，如果

$$\pi_1 = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix} \quad 和 \quad \pi_2 = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix},$$

那么

$$\pi_1^* = \begin{pmatrix} bb & br & rb & rr \\ bb & br & rb & rr \end{pmatrix} \quad 和 \quad \pi_2^* = \begin{pmatrix} bb & br & rb & rr \\ bb & rb & br & rr \end{pmatrix}. \quad (8.9)$$

注意，$\mathrm{Inv}(\pi_1^*)=4$，$\mathrm{Inv}(\pi_2^*)=2$，这使得 S^* 下的模式(等价类)数量是$\frac{1}{2}(4+2)=3$. 这与我们前面的观察模式是$\{bb\}$，$\{br, rb\}$和$\{rr\}$一致.

465

对于树着色(例 8.4)，注意一共存在 2^7 种树着色：树中的每一个顶点都可以得到两种颜色之一. 假设 π_i^* 是对应于图 8.18 所示的标签排列 π_i 的树着色排列. 写出 π_i^* 是不切实际的. 然而，注意到 $\pi_1^*=I^*$ 使所有 2^7 种树着色不变，所以有 $\mathrm{Inv}(\pi_1^*)=2^7=128$. 同样，排列 π_2 交换顶点 4 和 5. 因此 π_2^* 正好使顶点 4 和顶点 5 有相同颜色的着色不变，即使 2^6 种着色不变. 因此 $\mathrm{Inv}(\pi_2^*)=2^6=64$. 类似地，$\mathrm{Inv}(\pi_3^*)=2^6=64$，$\mathrm{Inv}(\pi_4^*)=2^4=16$，$\mathrm{Inv}(\pi_5^*)=2^5=32$，$\mathrm{Inv}(\pi_6^*)=2^3=8$，$\mathrm{Inv}(\pi_7^*)=2^3=8$，$\mathrm{Inv}(\pi_8^*)=2^4=16$. 因此，不同着色的数量或不同模式的数量是

$$\frac{1}{8}(128+64+64+16+32+8+8+16)=42.$$

8.4.3 自同构下的图着色等价

假设我们希望使用绿色(G)、黑色(B)和白色(W)着色图 8.15 中的图 $K_{1,3}$的顶点，而且不要求由一条边连接的两个顶点得到不同的颜色. 我们认为这样的两个着色等价，如果一种着色可以从另一种着色通过自同构而得到. 图 8.15 的顶点 1 得到绿色而其他顶点得到各不相同的颜色的任意两个着色是等价的，因为使用旋转 120°或 240°或相对于边{1, 2}，{1, 3}和{1, 4}的反射可以把一种这样的着色映射成另一种着色. 这些着色如图 8.19 所示. 旋转也表明顶点 1 得到 G，其余有两个顶点得到 B，最后一个顶点得到 W 的所有着色是等价的. 这些着色如图 8.20 所示. 类似地，对于{G, B, W}中的任意两种不同颜色的选择，$X\neq Y$，且 X 和 Y 都不等于 G，存在一个模式(等价着色的集合)，其中，顶点 1 得到 G，其余的两个顶点首先从$\{X, Y\}$中选择颜色 X，最后的顶点从$\{X, Y\}$中选择另一种颜色 Y. 一共有$3\times2=6$ 种方法选择两种不同颜色 X 和 Y，所以有 6 种这样的模式(等价类). 还存下面三种由一种着色组成的着色模式(等价类)：顶点 1 得到颜色 G，其他顶点都得到相同的颜色(也可能是 G). 因此，一共存在 10 种顶点 1 得到 G 的着色等价类. 对于顶点 1 的其他两种着色的选择，重复上面的计算，我们发现在自同构下一共有 30 种模式(等价类).

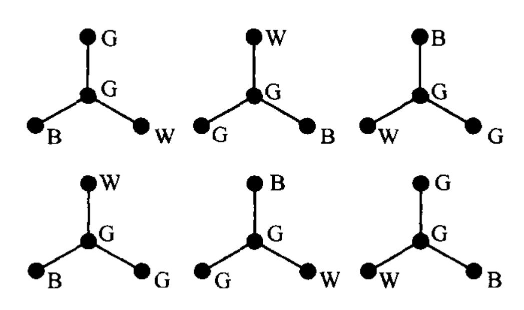


图 8.19　图 8.15 的顶点 1 得到颜色 G，而顶点 2，3 和 4 得到{G, B, W}的不同颜色的 $K_{1,3}$的着色模式(等价类)

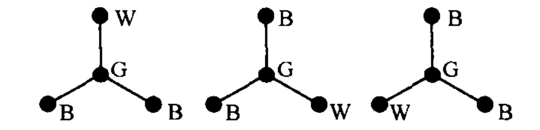


图 8.20　图 8.15 的顶点 1 得到颜色 G，而顶点 2，3 和 4 中，两个得到 B，另外一个得到 W 的 $K_{1,3}$的着色模式(等价类)

我们还可以根据伯恩赛德引理得到这一结果. 总之，$K_{1,3}$的顶点着色一共有 3^4 种. 假设 π_i^* 是对应于 $K_{1,3}$ 的自同构 π_i 的图着色排列. 恒等自同构 π_1 使所有这样的着色不变，所以 $|\mathrm{Inv}(\pi_1^*)|=3^4$. 相对于边{1, 2}的反射 π_2 使顶点 3 和顶点 4 得到相同颜色的着色不变. 一共有 3^3 种这样的着色，所以 $|\mathrm{Inv}(\pi_2^*)|=3^3$. 类似地，如果 π_3 和 π_4 分别是相对于边{1, 3}和{1, 4}的反射，那么有 $|\mathrm{Inv}(\pi_3^*)|=|\mathrm{Inv}(\pi_4^*)|=3^3$. 120°的旋转 π_5 使得顶点 2，3 和 4 得到相同颜色的着色不变. 因此，$|\mathrm{Inv}(\pi_5^*)|=3^2$. 类似地，对于 π_6 有 $|\mathrm{Inv}(\pi_6^*)|=3^2$，这是 240°

的旋转．一共有 6 个排列 π_i^*，所以伯恩赛德引理表明在自同构下，着色模式(等价类)的数量是

$$\frac{1}{6}[3^4+3^3+3^3+3^3+3^2+3^2]=\frac{1}{6}[180]=30.$$

8.4.4　开关函数的情况[⊖]

现在，我们把上面的着色理论运用于例 8.3 的开关函数．如果存在两个变量，那么我们认为两个这样的开关函数 T 和 U 相同，如果有 $T=U$，或者 $T(x_1x_2)=U(x_2x_1)$．如下扩展这一想法：两个 n 变量的开关函数 T 和 U 被认为相同，如果存在 $\{1, 2, \cdots, n\}$ 的一个排列 π 使得

$$T(x_1x_2\cdots x_n)=U(x_{\pi(1)}x_{\pi(2)}\cdots x_{\pi(n)}). \tag{8.10}$$

当 $n=2$ 时，两种可能的 π 是

$$\begin{pmatrix}1&2\\1&2\end{pmatrix}\quad 和 \quad\begin{pmatrix}1&2\\2&1\end{pmatrix}.$$

466～467

如果 $n=3$，对于(8.11)式所示的 π，满足(8.10)式的两个开关函数的例子由表 8.2 给出．

$$\pi=\begin{pmatrix}1&2&3\\2&3&1\end{pmatrix} \tag{8.11}$$

表 8.2　对于(8.11)式所示的 π，满足(8.10)式的开关函数 T 和 U

位串 x	$T(x)$	$U(x)$
000	1	1
001	0	0
010	1	0
011	1	0
100	0	1
101	0	1
110	1	1
111	0	0

对应于相同关系或等价关系，(8.10)式有意义．因为如果(8.10)式成立，那么可以通过为 U 设计的回路得到 T 的回路设计，其回路形式类似于图 8.2 中的回路．对于计算机工程，另一个相同关系也有意义．我们将在练习中加以讨论．

这一相同关系是如何适合于我们开发的形式结构呢？这里，D 是长度为 n 的位串的集合 B_n．设 π 是 $\{1, 2, \cdots, n\}$ 的任意排列，S_n 是 $\{1, 2, \cdots, n\}$ 的所有排列的群．那么一个位串 $x_1x_2\cdots x_n$ 可以看成是利用颜色 0 和 1 的 $\{1, 2, \cdots, n\}$ 的一个着色．与此对应的着色排列群是 S_n^*．这就是我们理论中的群 G．群 G^* 是群 $(S_n^*)^*$．$G^*=(S_n^*)^*$ 是由 S_n 中所有 π 的排列 $(\pi^*)^*$ 组成的．而 $(\pi^*)^*$ 是如何工作的呢？首先，注意 $\pi^*(x_1x_2\cdots x_n)=x_{\pi(1)}x_{\pi(2)}\cdots x_{\pi(n)}$．注意，$S_n^*$ 是位串集合的排列的群．$(S_n^*)^*$ 是由位串的着色的排列组成的．但是，使用颜色 0，1 的位串着色是一个开关函数．注意，根据定义，如果 U 是一个开关函数，那么有

$$\begin{aligned}[(\pi^*)^*U](x_1x_2\cdots x_n)&=U[\pi^*(x_1x_2\cdots x_n)]\\&=U(x_{\pi(1)}x_{\pi(2)}\cdots x_{\pi(n)}).\end{aligned}$$

因此，如果 $T=(\pi^*)^*U$，那么(8.10)式成立．

设 $n=2$，那么 $S_n=\{\pi_1, \pi_2\}$，其中

$$\pi_1=\begin{pmatrix}1&2\\1&2\end{pmatrix}\quad 和 \quad \pi_2=\begin{pmatrix}1&2\\2&1\end{pmatrix}.$$

$G=S_n^*$ 中的排列是

$$\pi_1^*=\begin{pmatrix}00&01&10&11\\00&01&10&11\end{pmatrix}\quad 和 \quad \pi_2^*=\begin{pmatrix}00&01&10&11\\00&10&01&11\end{pmatrix}. \tag{8.12}$$

468

一共有 $2^{2^2}=16$ 个两变量开关函数．这些函数是表 8.3 所示的 $T_1, T_2, \cdots, T_{16}$．这时，$G^*=(S_n^*)^*$ 中的排列是

[⊖] 本小节可以跳过．

$$(\pi_1^*)^* = \begin{pmatrix} T_1 & T_2 & T_3 & T_4 & T_5 & T_6 & T_7 & T_8 & T_9 & T_{10} & T_{11} & T_{12} & T_{13} & T_{14} & T_{15} & T_{16} \\ T_1 & T_2 & T_3 & T_4 & T_5 & T_6 & T_7 & T_8 & T_9 & T_{10} & T_{11} & T_{12} & T_{13} & T_{14} & T_{15} & T_{16} \end{pmatrix} \quad (8.13)$$

和

$$(\pi_2^*)^* = \begin{pmatrix} T_1 & T_2 & T_3 & T_4 & T_5 & T_6 & T_7 & T_8 & T_9 & T_{10} & T_{11} & T_{12} & T_{13} & T_{14} & T_{15} & T_{16} \\ T_1 & T_2 & T_5 & T_6 & T_3 & T_4 & T_7 & T_8 & T_9 & T_{10} & T_{13} & T_{14} & T_{11} & T_{12} & T_{15} & T_{16} \end{pmatrix} \quad (8.14)$$

注意，$(\pi_2^*)^*U$ 是函数 T，它作用于 01 的结果等于 U 作用于 10 的结果，它作用于 10 的结果等于 U 作用于 01 的结果，而对其余自变量它与 U 一致. 我们有 $\text{Inv}((\pi_1^*)^*)=16$ 且 $\text{Inv}((\pi_2^*)^*)=8$，两变量开关函数的模式或等价类的数量是 $\frac{1}{2}(16+8)=12$. 类似地，可以证明，三变量开关函数的模式数量是 80，四变量开关函数的模式数量是 3984，五变量开关函数的模式数量是 37 333 248(参见 Harrison[1965]或 Prather[1976]). 通过允许开关函数的其他对称变换(诸如交换值域或定义域内的 0 和 1)，如参见练习 23 和练习 24，我们能够进一步减少等价类的数量. 事实上，如果 $n=4$，这个数量可以减小到 222(Harrison[1965]和 Stone[1973]). 对于 $n=4$，这给出足够小的数，使得为获得含有各种等价类表达式的开关函数，准备一组最优回路是合理的.

表 8.3 16 个两变量开关函数

位串	$T(x)$															
x	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}	T_{13}	T_{14}	T_{15}	T_{16}
00	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
01	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
10	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
11	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

本节练习

1. 假设 $D=\{a, b, c\}$ 而 $R=\{1, 2\}$. 确定 $C(D, R)$ 中的所有着色.
2. 如果允许的颜色集合是{红，绿，蓝}，那么对于一个立方体的顶点，有多少种着色(不必不同)?
3. 如果有 6 种颜色可用，那么正四面体的顶点的允许着色(不必不同)有多少种?
4. 对于例 8.1，通过计算下列各式，验证(8.5)式、(8.7)式和(8.8)式：

[469]

   (a)$(\pi_2^* C_4)(3)$　　(b)$(\pi_3^* C_5)(2)$　　(c)$(\pi_4^* C_{11})(4)$
5. 对于例 8.2，通过计算下列各式，验证(8.9)式：
   (a)$(\pi_2^* br)(1)$　　(b)$(\pi_2^* br)(2)$　　(c)$(\pi_2^* rr)(1)$
6. 假设 $D=\{1, 2, 3, 4\}$，$R=\{1, 2\}$，而 G 是由等式(8.3)中的排列组成的.
   (a)假设 f 和 g 是下面的着色. 对于所有的 a，有 $f(a)=1$，且假设 $g(1)=g(2)=2$，$g(3)=g(4)=1$. fS^*g 成立吗?
   (b)假设 $f(1)=f(3)=2$，$f(2)=f(4)=1$，$g(1)=g(2)=2$，$g(3)=g(4)=1$. fS^*g 成立码?
   (c)确定 π_2^*.　　(d)确定 π_3^*.　　(e)确定 π_4^*.
   (f)确定 $\text{Inv}(\pi_2^*)$.　　(g)确定 $\text{Inv}(\pi_3^*)$.　　(h)确定 $\text{Inv}(\pi_4^*)$.
   (i)确定 S.　　(j)确定 S^*.
7. 假设 G 是由下面的排列组成的，重复练习 6[(e)和(h)除外]：
$$\pi_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}, \quad \pi_2 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 1 & 4 \end{pmatrix}, \quad \pi_3 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 2 & 4 \end{pmatrix}.$$
8. 在例 8.2 中，假设 $k=2$，而且有三种颜色可用：红(r)，蓝(b)和紫(p). 设 π_2 是如下所示的排列，确定 π_2^*.

$$\pi_2 = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}.$$

9. 在例 8.2 中，假设 $k=3$ 而且有二种颜色可用：红(r)，蓝(b).
   (a)确定 G^*.
   (b)使用伯恩赛德引理确定不同项链的数量.
   (c)通过枚举不同项链，验证你的答案.
10. 在 8.1 节练习 15 中，确定：
    (a)D　　(b)R　　(c)G
    (d)G^*　　(e)不同着色的数量.
11. 如果使用{G，B，W}中的颜色给图 8.15 中的图 $K_{1,3}$ 着色，确定下列模式：
    (a)顶点 1 着色成 W，顶点 2，3 和 4 分别着色成 G，B，B 的模式.
    (b)所有顶点都着色成 W 的模式.
12. 如果使用{G，B，W，P}中的颜色给图 $K_{1,3}$ 着色，确定模式的数量：
    (a)通过描述这些模式.　　(b)利用伯恩赛德引理.
13. 如果使用{G，B，W}中的颜色给图 Z_4 着色，描述所有模式并使用伯恩赛德引理计数这些模式的数量.
14. 如果使用{B，W}中的颜色给图 8.16 中的图 K_4-K_2 着色，描述所有模式并使用伯恩赛德引理计数这些模式的数量.　470
15. 对于例 8.4，证实：
    (a)$\text{Inv}(\pi_3^*)=2^6$　　(b)$\text{Inv}(\pi_4^*)=2^4$
    (c)$\text{Inv}(\pi_5^*)=2^5$　　(d)$\text{Inv}(\pi_6^*)=2^3$
16. 对于例 8.3，通过计算下列各式验证(8.14)式：
    (a)$(\pi_2^*)^*T_3$　　(b)$(\pi_2^*)^*T_{12}$　　(c)$(\pi_2^*)^*T_{15}$
17. 对于 8.1 节练习 9 的情况.
    (a)确定 G^*.
    (b)使用伯恩赛德引理计算不同着色的数量.
    (c)通过把你在 8.1 节中的答案与你给出的等价类枚举做比较，来验证你的答案.
18. 确定通过 0°或 180°的旋转 4×4 阵列的 2 着色的不同方法数量.
19. 证明图 8.18 中的 8 个排列定义一个群.
20. 假设我们考虑{1，2，3}上的一个弱序，把{1，2，3}的每一个元素着色成深色或浅色. 这时，我们区别图 8.21 中的弱序 A 和弱序 B. 如果我们使用例 8.6 的相同概念(与着色无关). 计数不同的已着色的弱序的数量.

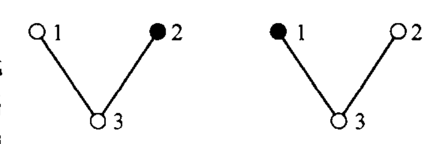
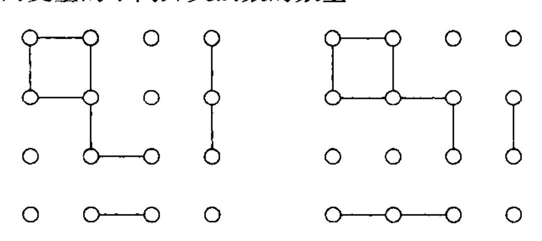


图 8.21　被着色的弱序

21. 假设 $V=\{1, 2, \cdots, p\}$. 回想一下，顶点集合 V 上的图(V, E)与给 V 的每一个 2 子集赋值 0 或 1 的函数之间存在一一对应. 其想法是：
$$f(\{i,j\}) = 1 \quad \text{当且仅当} \quad \{i,j\} \in E.$$
    函数 f 是使用颜色 0 和 1 对 V 的所有 2 子集的集合 D 的一个着色.
    (a)如果 $p=3$，确定所有这样的函数和它们对应的图.
    (b)如果 H 和 H' 是 V 上的两个图，而 f 和 f' 是它们对应的函数，证明 H 和 H' 同构当且仅当存在关于 D 的排列 π，使得对于 D 中的所有$\{i, j\}$，有 $f(\{i, j\})=f'(\pi(\{i, j\}))$，即使得 f 和 f' 等价.
    (c)设 G 是 D 的所有排列 π 的群. 如果 $p=3$，写出 G 的所有元素，并对于 G 中所有 π_i 计算 $\text{Inv}(\pi_i^*)$.
    (d)利用伯恩赛德引理确定三个顶点的不同(非同构)图的数量，并通过确定等价(同构)图的类来证实你的答案.　471
22. 对于 $p=4$，重复练习 21 的(a)、(c)和(d).

23. 假设 D 是长度为 3 的所有位串的集合，并设 G 是由恒等排列和通过交换 0 和 1 对位串求补的排列组成的群(参见 8.1 节练习 14). 确定不同开关函数的数量(使用颜色 0 和 1 的不同着色数量)，即 G 衍生的等价关系下的等价类的数量. 注意，在这种情况下，我们没有 S_n.
24. 假设两个开关函数被认为是等价的，如果其中一个开关函数可以从另一个开关函数通过排列或如练习 23 中那样对变量求补，或同时使用两种方法得到. 确定两变量的不同开关函数的数量.
25. (Reingold，Nievergelt 和 Deo[1977])一个集成电路制造商制作由排列成 4×4 阵列的 16 个元素组成的芯片. 这些元素与水平或垂直方向相邻的某些元素相连接. 图 8.22 给出若干样本相互连接模式. 这样的连接模式的光掩膜被用于在芯片上沉积出连接. 两个模式被认为是相同的，如果同一个光掩膜可以用于这两个模式. 例如，通过绕对角线翻转光掩膜，图 8.22 所示的两个连接模式可以使用一个光掩膜. 因此，它们被认为是相同的. 为了布局所有可能的连接模式需要多少个光掩膜? 通过定义适当的 D，R 和 G，把这一问题阐述成一个着色问题. 然而，你不要企盼计算 G^* 或利用目前开发的工具完全解决这一问题.


图 8.22 芯片的样本相互连接模式

8.5 循环指标

8.5.1 作为循环积的排列

把伯恩赛德引理运用于很多计数问题会变得相当麻烦. 例如，计算群 G^* 中的排列会变得很长、很复杂. 我们将开发另一个过程，在这一过程中我们得到的信息比伯恩赛德引理提供的更多.

排列

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 1 \end{pmatrix}$$

472

在数之间循环，它把 1 传递给 2，把 2 传递给 3，把 3 传递给 4，把 4 传递给 5，把 5 传递给 1. 这一排列可以简写成(12345). 更一般地，$(a_1a_2\cdots a_{m-1}a_m)$表示这样的排列：把 a_1 变成 a_2，把 a_2 变成 a_3，…，把 a_{m-1}变成 a_m，把 a_m 变成 a_1. 我们称这一排列为循环排列(cyclic permutation). 例如，排列

$$\begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}$$

就是循环排列(132). 现在考虑下面的排列：

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 5 & 1 & 6 & 3 & 2 & 4 \end{pmatrix}.$$

这个排列是由两个循环排列(152)和(364)组成的. 我们说上面的排列是这两个循环的积(product)，写作(152)(364). 如果我们把(152)这样的循环当作是保持 3，4 和 6 不变，那么上面的两个循环的积相当于两个排列的积. 同样地，排列$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 1 & 3 & 5 & 6 & 4 \end{pmatrix}$是三个循环的积. (12)(3)(456)，其中(3)意味着 3 被映射到自身. 在这一积中，三个循环在其中任意两个循环都没有相同元素的意义下是不相交的(disjoint).

现在我们要证明{1，2，…，n}的每一个排列都可以写作不相交的循环的积，其中 1，2，…，n 中的每一个元素在某个循环中出现. 为了弄明白其道理，我们取下面的排列：

$$\begin{pmatrix}1&2&3&4&5&6&7&8\\5&4&8&6&7&2&1&3\end{pmatrix}.$$

取元素 1. 它变到 5，依次地 5 变到 7，7 变到 1，所以我们有一个循环(157). 取不在这个循环中的第一个元素 2，它变成 4，4 变成 6，而 6 变成 2，所以我们有循环(246). 取不出现在前面的循环中的第一个元素 3，它变成 8，而 8 变成 3，所以我们有循环(38). 因此原来的排列可以写成三个不相交的循环的积(157)(246)(38). 对任意的排列都可以进行类似的推理.

排列 π 能否用两种不同的方法写成不相交的循环(每一个元素都在某一个循环里)的积呢？如果我们认为$(a_1a_2\cdots a_m)$和$(a_ia_{i+1}\cdots a_ma_1a_2\cdots a_{i-1})$是不相同的，那么上面问题的答案是肯定的. 然而，我们认为它们是相同的，因为它们对应于相同的排列. 因此，(123)和(231)，(321)都是相同的. 另外，我们书写循环的顺序与不相交循环的积无关. 例如，我们认为(12)(345)和(345)(12)是相同的. 假设

$$(xyz\cdots)\cdots(abc\cdots)=\pi=(uvw\cdots)\cdots(\alpha\beta\gamma\cdots).$$

473

如果写出排列的两种方法是不同的，那么一定存在一个数 k 使得左边包含 k 的循环不同于右边包含 k 的循环. 我们可以在这两个循环中把 k 写在第一位. 于是无论 π 把 k 变成什么，它都一定在每个循环的第二个位置上，而无论 π 再把这个元素变成什么，它都一定在每个循环中第三个位置上，以此类推. 因此，这两个循环一定相同. 概括上述说明，我们有下面的结果.

定理 8.5 *$\{1, 2, \cdots, n\}$的每一个排列都可以正好用一种方法写成不相交循环的积且$\{1, 2, \cdots, n\}$的每一个元素出现在某个循环中.*

我们将把定理 8.5 中所陈述的书写排列的唯一方法称为这个排列的循环分解(cycle decomposition).

8.5.2 波利亚定理的特殊情况

现在我们已准备好给出另一个计数等价类的结果，这一结果是我们一直讨论的主定理的一个特殊情况. 假设 $\text{cyc}(\pi)$计数排列 π 的唯一循环分解中循环的数量. 例如，如果 $\pi=(12)(3)(456)$，那么 $\text{cyc}(\pi)=3$.

定理 8.6(波利亚定理的特殊情况) *假设 G 是集合 D 的排列群，且 $C(D, R)$是使用 R 中的颜色的 D 中元素的着色集合，其中 R 是一个 m 元素集合. 那么 $C(D, R)$中不同着色的数量(G^*衍生的等价关系 S^* 下的等价类或模式的数量)由下式给出：*

$$\frac{1}{|G|}\left[m^{\text{cyc}(\pi_1)}+m^{\text{cyc}(\pi_2)}+\cdots+m^{\text{cyc}(\pi_k)}\right],$$

其中 $G=\{\pi_1, \pi_2, \cdots, \pi_k\}$.

注意，这一定理允许我们不用先计算 G^* 就可计算不同着色的数量. 我们在 8.5.6 节给出它的证明.

为了说明这一定理，我们再一次考虑例 8.1 的 2×2 阵列的情况. G 中有四个排列，即如图 8.13 所示的 4 个旋转 π_1，π_2，π_3，π_4. 这些排列有如下循环分解：$\pi_1=(1)(2)(3)(4)$，$\pi_2=(1432)$，$\pi_3=(13)(24)$，$\pi_4=(1234)$. 因此，$\text{cyc}(\pi_1)=4$，$\text{cyc}(\pi_2)=1$，$\text{cyc}(\pi_3)=2$，$\text{cyc}(\pi_4)=1$. 不同着色数量(模式数量)是$\frac{1}{4}(2^4+2^1+2^2+2^1)=6$，这与我们前面的计算一致.

对于例 8.2，设有 k 个珠子的项链，我们可以把 G 中的排列 π_1 写成下面的形式：

$$\pi_1=(1)(2)\cdots(k),$$

而且，如果 k 是偶数，则 π_2 可以写成如下形式：

$$\pi_2=(1k)(2k-1)(3k-2)\cdots\left(\frac{k}{2}\ \frac{k}{2}+1\right),$$

474

如果 k 是奇数，则 π_2 可以写成如下形式：

$$\pi_2=(1k)(2k-1)(3k-2)\cdots\left(\frac{k-1}{2}\ \frac{k+3}{2}\right)\left(\frac{k+1}{2}\right).$$

例如，如果 $k=4$，则 $\pi_2=(14)(23)$．如果 $k=5$，则 $\pi_2=(15)(24)(3)$．于是有，$\mathrm{cyc}(\pi_1)=k$，如果 k 是偶数，则 $\mathrm{cyc}(\pi_2)=k/2$，如果 k 是奇数，则 $\mathrm{cyc}(\pi_2)=(k+1)/2$．因此，如果 k 是偶数，则不同项链的数量为 $\frac{1}{2}(2^k+2^{k/2})$，如果 k 是奇数，则不同项链的数量为 $\frac{1}{2}(2^k+2^{(k+1)/2})$．例如，当 $k=2$ 时，结果为 3，这与我们前面的计算一致．当 $k=3$ 时，结果为 6，这留给读者进行验证(练习 8)．如果存在 3 种不同的珠子颜色且 k 是偶数，那么我们有 $\frac{1}{2}(3^k+3^{k/2})$ 个不同项链．例如，$k=2$，我们有 6 个不同项链．如果珠子的颜色是 r，b 和 p，那么这 6 个等价类是 $\{rb, br\}$，$\{rp, pr\}$，$\{bp, pb\}$，$\{rr\}$，$\{bb\}$，$\{pp\}$．

对于例 8.4 中的树着色，G 由图 8.18 中的 8 个排列 π_1，π_2，…，π_8 给定．我们有下面的结果：

$$\begin{aligned}
&\pi_1=(1)(2)(3)(4)(5)(6)(7), && \pi_2=(1)(2)(3)(45)(6)(7),\\
&\pi_3=(1)(2)(3)(4)(5)(67), && \pi_4=(1)(23)(46)(57),\\
&\pi_5=(1)(2)(3)(45)(67), && \pi_6=(1)(23)(4756),\\
&\pi_7=(1)(23)(4657), && \pi_8=(1)(23)(47)(56).
\end{aligned}$$

因此，不同的树着色数量是

$$\frac{1}{8}(2^7+2^6+2^6+2^4+2^5+2^3+2^3+2^4)=42,$$

这正是我们在前面已计算的结果．

8.5.3　再论自同构下的图着色等价

让我们回到 8.4.3 节中讨论的图 $K_{1,3}$ 的顶点着色问题．容易看到，有下面的结果：

$$\begin{aligned}
&\pi_1^*=(1)(2)(3)(4), && \pi_2^*=(1)(2)(3\ 4), && \pi_3^*=(1)(3)(2\ 4),\\
&\pi_4^*=(1)(4)(2\ 3), && \pi_5^*=(1)(3\ 4\ 2), && \pi_6^*=(1)(4\ 2\ 3).
\end{aligned}$$

因此，如果我们使用颜色 G，B，W 着色，那么不同着色的数量由下式给出：

$$\begin{aligned}
&\frac{1}{6}\left[3^{\mathrm{cyc}(\pi_1^*)}+3^{\mathrm{cyc}(\pi_2^*)}+3^{\mathrm{cyc}(\pi_3^*)}+3^{\mathrm{cyc}(\pi_4^*)}+3^{\mathrm{cyc}(\pi_5^*)}+3^{\mathrm{cyc}(\pi_6^*)}\right]\\
&=\frac{1}{6}\left[3^4+3^3+3^3+3^3+3^2+3^2\right]=30.
\end{aligned}$$

475　这与我们前面的计算一致．

8.5.4　开关函数的情况[⊖]

对于例 8.3 中的开关函数，我们需要考虑 $G=H^*$ 的元素．当我们考虑两变量开关函数时，有 $G=\{\pi_1^*, \pi_2^*\}$，其中 π_1 和 π_2 如下所示：

$$\pi_1=\begin{pmatrix}1&2\\1&2\end{pmatrix}\quad 和 \quad \pi_2=\begin{pmatrix}1&2\\2&1\end{pmatrix}.$$

根据 8.4.4 节中的计算，π_1^* 和 π_2^* 由(8.12)式给出．现在，我们可以认为 π_1^* 和 π_2^* 是作用于$\{1, 2, 3, 4\}$上，在这种情况下，$\pi_1^*=(1)(2)(3)(4)$，而 $\pi_2^*=(1)(23)(4)$，所以 $\mathrm{cyc}(\pi_1^*)=4$，

⊖　本小节可以跳过．

$\text{cyc}(\pi_2^*)=3$，且不同的开关函数的数量是$\frac{1}{2}(2^4+2^3)=12$，这与我们前面的计算一致.

8.5.5 排列群的循环指标

用类似于生成函数的方式概括排列群中排列的循环结构是有益的. 假设π是一个排列，而且在它的唯一循环分解中有b_1个长度为1的循环，b_2个长度为2的循环，⋯. 那么，如果x_1，x_2，⋯是占位符，而且k至少是π的循环分解中最长的循环的长度，那么我们可以使用表达式$x_1^{b_1}x_2^{b_2}\cdots x_k^{b_k}$来编码$\pi$. 另外，我们可以通过对$G$的成员的这些表达式取和，再除以$G$的排列数量来编码整个排列群$G$. 即如果$k$是$G$中任意$\pi$的循环分解[⊖]的最长循环的长度，那么我们定义

$$P_G(x_1,x_2,\cdots,x_k)=\frac{1}{|G|}\sum_{\pi\in G}x_1^{b_1}x_2^{b_2}\cdots x_k^{b_k},$$

且称$P_G(x_1, x_2, \cdots, x_k)$为$G$的*循环指标*(cycle index). 例如，考虑例8.4. 于是，使用图8.18的表记法，$\pi_6=(1)(23)(4756)$，而它的对应编码是$x_1x_2x_4$. 还有，$\pi_4=(1)(23)(46)(57)$，它被编码为$x_1x_2^3$. 通过类似的分析，这一排列群的循环指标是

$$\begin{aligned}P_G(x_1,x_2,\cdots,x_8)=\frac{1}{8}[&x_1^7+x_1^5x_2+x_1^5x_2+x_1x_2^3\\&+x_1^3x_2^2+x_1x_2x_4+x_1x_2x_4+x_1x_2^3].\end{aligned}\tag{8.15}$$

注意，如果π是对应编码为$x_1^{b_1}x_2^{b_2}\cdots x_k^{b_k}$的一个排列，那么$\text{cyc}(\pi)=b_1+b_2+\cdots+b_k$，而且如果取所有$x_i$为$m$，那么$x_1^{b_1}x_2^{b_2}\cdots x_k^{b_k}$是$m^{\text{cyc}(\pi)}$. 因此，可以如下重述定理8.6：

推论8.6.1 *假设G是集合D的排列群，$C(D, R)$是使用R中颜色的D元素的着色的集合，而R是m个元素的集合. 那么$C(D, R)$中的不同着色的数量由$P_G(m, m, \cdots, m)$给出.* [476]

我们将在8.6节扩展这一结果.

为了使用这个结果，我们回到例8.4. 这里，$m=2$. 在(8.15)式中，令$x_1=x_2=\cdots=x_8=2$,得到

$$P_G(2,2,\cdots,2)=\frac{1}{8}(2^7+2^6+2^6+2^4+2^5+2^3+2^3+2^4)=42,$$

这与我们前面得到的不同着色的数量一致.

现在，我们考虑例8.5. 在8.1节，我们确认了四面体的12种不同的对称. 这些对称可以认为是图8.5中的字母a，b，c，d的排列. 恒等对称是排列$(a)(b)(c)(d)$，这一排列有循环结构代码x_1^4. 绕连接顶点a与由b，c和d所决定的面的中心连线的120°旋转对应于排列$(a)(bdc)$，这个排列可以编码为x_1x_3(参见图8.6). 所有8个120°和240°的旋转对称有类似的结构和编码. 最后，绕连接边ab和边cd的中点连线的180°旋转对应于排列$(ab)(cd)$，这一排列的编码是x_2^2. 其他两个180°旋转有类似的编码. 因此循环指标由下式给出：

$$P_G(x_1,x_2,x_3)=\frac{1}{12}(x_1^4+8x_1x_3+3x_2^2).$$

我们寻找使用集合$R=\{CH_3, C_2H_5, H, Cl\}$中的四种颜色的集合$D=\{a, b, c, d\}$的着色. 因此，$m=4$，且不同着色的数量(不同分子的数量)由下面的计算给出：

$$P_G(m,m,m)=\frac{1}{12}[4^4+8(4)(4)+3(4)^2]=36.$$

⊖ 也可以取$k=|D|$，并注意到G中任意π的循环分解中最长循环的长度最多是k.

8.5.6　定理 8.6 的证明[⊖]

为了证明定理 8.6，我们把伯恩赛德引理(定理 8.4)运用于 G^*. 因为 $|G|=|G^*|$，只要证明 $m^{cyc(\pi)}=Inv(\pi^*)$即可. 设 π 在 G 中. 我们尝试计算 $Inv(\pi^*)$. 注意，经过 π^* $C(D, R)$中的元素保持不变，当且仅当对于 D 的对应排列 π，在 π 的每一个循环中，D 的所有元素得到相同的颜色. 例如，假设 $\pi=(12)(345)(67)(8)$. 设 f 是这样的着色：使得 $f(1)=f(2)=$黑色，$f(3)=f(4)=f(5)=$白色，$f(6)=f(7)=$红色，$f(8)=$蓝色. 那么显然 $\pi^* f$ 是与 f 相同的着色.

总之，为了确定一个经过 π^* 保持不变的着色，我们计算 π 的循环分解，并用相同的颜色着色一个循环中的每一个元素. 现在，π 在其循环分解中有 cyc(π)个不同的循环，且对于每一个循环的共同颜色，我们有 m 种选择. 因此，经过 π^* 保持不变的不同着色有 $m^{cyc(\pi)}$ 个. 简而言之，$Inv(\pi^*)=m^{cyc(\pi)}$.

477

本节练习

1. 确定 8.2 节练习 1 中的每一个排列的循环分解.
2. 确定 8.1 节练习 9 情况中出现的所有排列 π_i 的循环分解.
3. 计算 8.2 节练习 10 的(a)、(b)和(c)中的每一个排列的 cyc(π).
4. 对于 8.2 节练习 10 的(a)、(b)和(c)中的每一个排列，编码该排列为 $x_1^{b_1}x_2^{b_2}\cdots x_k^{b_k}$.
5. 对于 8.2 节练习 10 中的每一个排列群，计算循环指标.
6. 已知 D，R 和 G 如 8.4 节练习 6 所示：
   (a)通过定理 8.6 确定不同着色的数量.
   (b)使用引理 8.6.1 确定不同着色的数量.
7. 设 D，R 和 G 如 8.4 节练习 7 所示，重复练习 6.
8. 对于例 8.2，验证当 $k=3$ 时存在 6 种不同的项链.
9. 对于例 8.2，使用定理 8.6 确定不同的项链的数量. 如果：
   (a)颜色的数量是 2，而 k 是 4.
   (b)颜色的数量是 2，而 k 是 5.
   (c)颜色的数量是 3，而 k 是 3.
   (d)颜色的数量是 3，而 k 是 4.
10. 使用引理 8.6.1 重复练习 9.
11. (a)对于 8.1 节练习 9，使用定理 8.6 确定等价类的数量.
    (b)使用引理 8.6.1，重复(a).
12. (a)对于 8.1 节练习 15，使用定理 8.6 确定不同着色的数量.
    (b)使用引理 8.6.1，重复(a).
13. (a)对于 8.1 节练习 8，使用定理 8.6 确定不同着色的数量.
    (b)使用引理 8.6.1，重复(a).
14. 对于例 8.1，验证：对于每一个 π_i，$i=1, 2, 3, 4$，$Inv(\pi_i^*)=m^{cyc(\pi_i)}$.
15. 如果两个图着色的等价如 8.4.3 节所定义，继续 8.2 节的练习 23，确定下面情况不同着色的数量：
    (a)使用 2 种颜色的图 L_4 的着色.　　(b)使用 3 种颜色的图 L_4 的着色.
    (c)使用 2 种颜色的图 Z_4 的着色.　　(d)使用 3 种颜色的图 Z_4 的着色.
    (e)使用 2 种颜色的图 K_4-K_2 的着色.　　(f)使用 3 种颜色的图 K_4-K_2 的着色.
16. 使用本节的方法重复 8.4 节练习 20.
17. (a)使用定理 8.6，确定 $p=3$ 个顶点的非同构图的数量(参见 8.4 节练习 21).

⊖ 本小节可以跳过.

(b)使用引理 8.6.1，重复(a).

18. 对于 $p=4$，重复练习 17. 478

19. (a)对于 8.4 节练习 23，使用定理 8.6 确定不同开关函数的数量.

    (b)使用引理 8.6.1，重复(a).

20. 对于 8.4 节练习 24，重复练习 19.

21. 如果采纳 8.4 练习 24 中的相同的定义，确定 3 变量开关函数的数量，已知对应的排列群的循环指标由下式给出：

$$\frac{1}{12}[x_1^8+4x_2^4+2x_1^2x_3^2+2x_2x_6+3x_1^4x_2^2].$$

22. 考虑 3 维空间中的一个立方体. 它有 8 个顶点. 下面的对称对应于这些顶点的排列. 以 $x_1^{b_1}x_2^{b_2}\cdots x_k^{b_k}$ 的形式编码每一个对称，并计算对应于这些对称的所有排列群 G 的循环指标：

    (a)恒等对称.

    (b)绕连接相对平面的中心连线的 180°旋转(一共有三个).

    (c)绕连接相对平面的中心连线的 90°或 270°旋转(一共有六个).

    (d)绕连接对边中点连线的 180°旋转(一共有六个).

    (e)绕相对顶点连线的 120°旋转(一共有八个).

23. 对于练习 22，确定使用两种颜色，红色和蓝色，着色这个立方体的顶点的不同方法的数量.

24. 完成 8.4 节练习 25 的解.

25. 转置(transposition)是一个循环(ij). 证明每一个排列都是转置的积[提示：只要证明每一个循环都是转置的积即可].

26. 继续练习 25，把(123456)写成转置的积.

27. 把 8.2 节练习 1 中的每一个排列写成转置的积.

28. 证明可以用多种方法把一个排列写成转置的积.

29. 尽管排列可以用多种方法写成转置的积，但可以证明，把给定排列写成转置的积的形式时，每一种方法或者包含偶数个转置或包含奇数个置换(作为证明，参见练习 31). 因此，一个排列可以称为偶排列，如果把它写成转置的积的每一种方法都使用偶数个转置，否则它是奇排列.

    (a)确认{1，2，3}的所有偶排列.

    (b)证明$\{1, 2, \cdots, n\}$的偶排列集合组成一个群.

    (c)$\{1, 2, \cdots, n\}$的奇排列集合是否组成一个群?

30. 有代码 $x_1^{b_1}x_2^{b_2}\cdots x_k^{b_k}$ 的$\{1, 2, \cdots, n\}$的排列数量由下面的公式给出：

$$\frac{n!}{b_1!b_2!\cdots b_n!1^{b_1}2^{b_2}\cdots n^{b_n}}.$$

    这个公式称为柯西公式. 479

    (a)对于 $n=5$，$b_1=3$，$b_2=1$，$b_3=b_4=b_5=0$，验证这个公式.

    (b)对于 $n=3$ 和所有可能的代码，验证这个公式.

31. 假设

$$D_n=(2-1)(3-2)(3-1)(4-3)(4-2)(4-1)\cdots(n-1). \tag{8.16}$$

    如果 π 是$\{1, 2, \cdots, n\}$的一个排列，根据 D_n，通过用$(\pi(i)-\pi(j))$取代(8.16)式中的$(i-j)$来定义 πD_n.

    (a)确定 D_5.

    (b)如果 π 是如下所示的排列，确定 πD_5：

$$\pi=\begin{pmatrix}1 & 2 & 3 & 4 & 5\\ 4 & 3 & 5 & 2 & 1\end{pmatrix}.$$

    (c)证明：如果 π 是一个转置，那么 $\pi D_n=-D_n$.

(d)从(c)推出结论：如果 π 是偶数个转置的积，那么有 $\pi D_n = D_n$，如果 π 是奇数个转置的积，那么 $\pi D_n = -D_n$.

(e)证明：一个排列不可能既可以写成偶数个转置的积又可以写成奇数个转置的积.

8.6 波利亚定理

8.6.1 着色的目录

我们可能不仅对不同着色的数量感兴趣，而且对某一种类的着色的数量感兴趣. 例如，在例 8.1 中，我们也许对计数这个 2×2 阵列的正好使用两个黑色的不同的 2 着色数量感兴趣；在例 8.5 中，我们也许对计数至少有一个氢原子的不同分子的数量感兴趣等等. 现在我们给出回答这类问题的一个一般结果.

设 $D=\{a_1, a_2, \cdots, a_n\}$ 是要被着色的对象的集合，而 $R=\{r_1, r_2, \cdots, r_m\}$ 是颜色的集合. 我们通过给每一种颜色 r 指定一个权 $w(r)$ 来区分着色. 这个权既可以是一个符号也可以是一个数.

当我们给颜色赋予权之后，我们就可以赋*着色权*(weight to a coloring). 着色权定义为分配给 D 中元素的各颜色的权的积. 为了说明这一定义，假设 $R=\{x, y, z\}$，$w(x)=1$，$w(y)=5$，$w(z)=7$. 假设着色对象是图 8.18 中的第一棵二叉树的 7 个顶点，我们用颜色 x 着色顶点 1，2，4，6，用颜色 y 着色顶点 3，7，而用颜色 z 着色顶点 5. 于是，该着色权等于 $w(x)^4w(y)^2w(z)=(1)^4(5)^27=175$. 如果 $w(x)=r$，$w(y)=g$，$w(z)=b$，那么着色权等于 r^4g^2b. 我们将在下面看到着色权是如何非常有效地编码着色的.

现在，假设 K 是一个着色的集合，K 中的着色权的和称为 K 的*目录*(inventory). 例如，假设 $D=\{a, b, c, d\}$，$R=\{x, y, z\}$，而且 $w(x)=r$，$w(y)=g$，$w(z)=b$. 设 $C(D, R)$ 中的着色 f_1，f_2 和 f_3 的定义如下：

480

$$f_1(a)=x,\quad f_1(b)=y,\quad f_1(c)=y,\quad f_1(d)=z,$$
$$f_2(a)=z,\quad f_2(b)=z,\quad f_2(c)=x,\quad f_2(d)=z,$$
$$f_3(a)=x,\quad f_3(b)=z,\quad f_3(c)=y,\quad f_3(d)=x.$$

设 $W(f_i)$ 是着色 f_i 的权. 于是 $W(f_1)=w(x)w(y)w(y)w(z)=rg^2b$，类似地，$W(f_2)=rb^3$，$W(f_3)=r^2gb$. 集合 $K=\{f_1, f_2, f_3\}$ 的目录是 $rg^2b+rb^3+r^2gb$. 如果所有颜色的权的符号不同，那么着色权表示所用颜色的分布. 例如，$W(f_1)=rg^2b$，则表示 f_1 使用颜色 x 一次，使用颜色 y 两次，使用颜色 z 一次. 着色集合的目录概括了这一集合中不同着色的颜色分布. 这类似于生成函数.

现在，假设 G 是集合 D 的一个排列群，而 f 和 g 是 $C(D, R)$ 中两个等价的着色. 于是，如在 8.4.2 节所观察的那样，存在 G 中的排列 π 使得对于所有 D 中的 a，有 $g(a)=f(\pi(a))$，如果 $D=\{a_1, a_2, \cdots, a_n\}$，于是，

$$W(f)=w[f(a_1)]w[f(a_2)]\cdots w[f(a_n)] \tag{8.17}$$

及

$$W(g)=w[g(a_1)]w[g(a_2)]\cdots w[g(a_n)]. \tag{8.18}$$

因为 π 是排列，所以集合 $\{a_1, a_2, \cdots, a_n\}$ 与集合 $\{\pi(a_1), \pi(a_2), \cdots, \pi(a_n)\}$ 正好有相同的元素. 因此，(8.17)式意味着

$$W(f)=w[f(\pi(a_1))]w[(\pi(a_2))]\cdots w[f(\pi(a_n))]. \tag{8.19}$$

但是，因为 $g(a)=f(\pi(a))$，(8.18)式和(8.19)式表明 $W(f)=W(g)$. 这样，我们就证明了下面的定理.

定理 8.7 *如果着色 f 和 g 是等价的，那么它们有相同的权.*

作为这一定理的一个结果，我们可以说*着色的等价类的权*(weight of an equivalence class of

coloring)，或同样地，模式的权(weight of pattern)．这就是该类中任意着色的权．我们还可以说一个模式集合的目录(inventory of a set of patterns)，或一个等价类的集合的目录，或模式目录(pattern inventory)，它等于这个集合中的模式的权的和．例如，考虑例8.1中2×2阵列的着色．一共存在6种着色模式，如图8.10所示．设黑色的权是b，白色的权是w．于是，图8.10的类1的着色权是b^4，类2的所有着色的权是b^3w，类3的所有着色的权是b^2w^2，类4的所有着色的权是b^2w^2，类5的所有着色的权是bw^3，类6的所有着色的权是w^4．注意，两个不同等价类可以有相同的权．这时，模式目录是

$$b^4+b^3w+2b^2w^2+bw^3+w^4. \tag{8.20}$$

我们发现只有一个等价类使用4次黑色，一个等价类使用3次黑色和一次白色，两个等价类使用2次黑色和2次白色，等等．这些信息可以直接从这一模式目录得到．如果只想要确定模式的数量，那么我们可以如第5章处理生成函数那样，取所有的权为1．这里，在(8.20)式中设$b=w=1$给出结果6，这就是模式数量．如果我们想要确定不使用黑色的模式数量，可以设w(黑色)$=0$，而w(白色)$=1$，或等价地，在(8.20)式中设$b=0$而$w=1$．这个结果为1．即只存在1个不使用黑色的模式，这对应于模式目录中的w^4项．现在，我们将要寻找一种在不知道等价类的情况下，计算模式目录的方法． 481

8.6.2 计算模式目录

定理8.8(波利亚定理)⊖ *假设G是集合D上的排列群，而$C(D, R)$是使用R中颜色的D的所有着色的集合．如果w是关于R的一个权赋值，那么$C(D, R)$的着色的模式目录由下式给出：*

$$P_G\left(\sum_{r\in R}w(r),\sum_{r\in R}[w(r)]^2,\sum_{r\in R}[w(r)]^3,\cdots,\sum_{r\in R}[w(r)]^k\right),$$

其中$P_G(x_1, x_2, x_3, \cdots, x_k)$是循环指标．

注意，引理8.6.1是这一定理的特殊情况，其中，对于R中的所有r都有$w(r)=1$．为了说明这一定理，我们再一次回到例8.1中的2×2阵列．注意，G是由排列$\pi_1=(1)(2)(3)(4)$，$\pi_2=(1432)$，$\pi_3=(13)(24)$，$\pi_4=(1234)$组成的．因此，其循环指标是由下式给出：

$$P_G(x_1,x_2,x_3,x_4)=\frac{1}{4}(x_1^4+2x_4+x_2^2).$$

现在，让我们对黑色着色赋予权b，对白色着色赋予权w．于是，$R=\{$黑色，白色$\}$且

$$\sum_{r\in R}w(r)=b+w,\qquad \sum_{r\in R}[w(r)]^2=b^2+w^2,$$

$$\sum_{r\in R}[w(r)]^3=b^3+w^3,\quad \sum_{r\in R}[w(r)]^4=b^4+w^4.$$

根据波利亚定理，模式目录通过取上述的$P_G(x_1, x_2, x_3, x_4)$并用$\sum_{r\in R}w(r)$替换x_1，用$\sum_{r\in R}[w(r)]^2$替换x_2等等给出．因此，模式目录是

$$\frac{1}{4}[(b+w)^4+2(b^4+w^4)+(b^2+w^2)^2]. \tag{8.21}$$

482

通过使用二项式展开(定理2.7)，我们可以展开(8.21)式并得到(8.20)式，这就是我们在前面对

⊖ 波利亚的基础定理第一次出现于他的经典论文(Pólya[1937])中．这一结果已被Redfield[1927]预见，但是很少有人能够理解Redfield的结果，而Pólya不知道这些结果．波利亚定理的扩展形式可以在de Bruijn[1959]中找到，对这一结果的阐述可以参见Liu[1968]．波利亚理论的著名阐述可以在de Bruijn[1964]找到．对于其他阐述，参见Bogart[1999]、Brualdi[1999]或Tucker[1995]．

模式目录的说明.

假设我们允许以三种颜色着色 2×2 阵列：黑色、白色和红色. 如果我们设 w(红色)$=r$，那么我们发现模式目录是

$$\frac{1}{4}[(b+w+r)^4+2(b^4+w^4+r^4)+(b^2+w^2+r^2)^2]=b^4+w^4+r^4+b^3w+w^3b$$

$$+b^3r+r^3b+w^3r+r^3w+2b^2w^2+2b^2r^2+2w^2r^2+3b^2wr+3w^2br+3r^2wb.$$

例如，我们看到有三个模式使用两次黑色、一次白色和一次红色. 这三个模式的例子如图 8.23 所示. 使用两次黑色，一次白色和一次红色的其他模式可以从其中一个模式通过旋转而得到. 而模式数量是在模式目录中取 $b=w=r=1$ 而得到的. 注意，一旦计算出循环指标，我们就能够很容易地运用它做大量不同的计数而不必重复指标的计算.

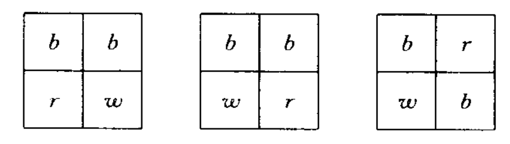


图 8.23 使用两次黑色、一次白色和一次红色的 2×2 阵列着色的三个不同模式的例子

接下来，考虑例 8.5 中的有机分子. 我们已经有下面的循环指标：

$$P_G(x_1,x_2,x_3)=\frac{1}{12}(x_1^4+8x_1x_3+3x_2^2).$$

假设我们想要确定至少包含一个氯原子的不同分子(模式)的数量. 比较容易的是先计算不包含氯原子的模式数量. 通过对每一种颜色 CH_3，C_2H_5 和 H 赋权 1，而对颜色 CL 赋权 0，我们就可以得到不包含氯原子的模式数量. 这时，对于所有的 $k\geqslant 1$，有

$$\sum_{r\in R}[w(r)]^k=[w(CH_3)]^k+[w(C_2H_5)]^k+[w(H)]^k+[w(Cl)]^k=1+1+1+0=3.$$

因此，模式目录由下式给出：$\frac{1}{12}(3^4+8(3)(3)+3(3)^2)=15$. 因为我们前面已经计算得到总共存在 36 个模式，所以，至少包含一个氯原子的模式数量是 $36-15=21$.

继续讨论本例，假设除 Cl 外对每一种颜色赋权 1，而对 Cl 赋权 c. 那么模式目录由下式给出：

$$\frac{1}{12}[(c+3)^4+8(c+3)(c^3+3)+3(c^3+3)^2]=c^4+3c^3+6c^2+11c+15.$$

我们得出结论：存在 1 个由四个氯原子组成的模式，3 个含三个氯原子的模式，6 个含两个氯原子的模式，11 个含一个氯原子的模式，以及 15 个不含氯原子的模式.

483

接下来，我们返回 8.4.4 节和 8.5.4 节所讨论的使用集合{G，B，W}中的颜色的 $K_{1,3}$的图着色问题. 有多少种着色不使用 G 呢? 排列 π_i^* 中的最长循环的长度是 3，根据 8.5.4 节的结果，我们看到

$$P_G(x_1,x_2,x_3)=\frac{1}{6}(x_1^4+3x_1^2x_2+2x_1x_3).$$

设 $w(G)=g$，$w(B)=b$，$w(W)=w$，我们得到

$$P_G(g,b,w)=\frac{1}{6}[(g+b+w)^4+3(g+b+w)^2(g^2+b^2+w^2)$$

$$+2(g+b+w)(g^3+b^3+w^3)]. \tag{8.22}$$

我们可以通过在(8.22)式中设 $g=0$，$b=w=1$ 来回答我们的问题，得到

$$\frac{1}{6}[16+24+8]=8.$$

因此，不使用 G 的 $K_{1,3}$ 的不同着色有 8 种. 读者应该验证这一结果. 如果想要知道有多少种着色正好使用两个 W 时，该怎么办？我们可以在(8.22)式中设 $g=b=1$，并计算模式目录：

$$\frac{1}{6}[(2+w)^4+3(2+w)^2(2+w^2)+2(2+w)(2+w^3)].$$

对上式进行化简，我们得到

$$8+10w+7w^2+4w^3+w^4. \tag{8.23}$$

有两个 W 的不同着色的数量由 w^2 的系数给出，即是 7. 读者应该验证这个结果.

8.6.3 开关函数的情况[⊖]

接下来转向例 8.3 的开关函数，这里取 $n=2$. 于是，G 是由(8.21)式给出的排列 π_1^* 和 π_2^* 组成的. 同前面一样，可以很自然地认为 π_1^* 是(1)(2)(3)(4)而 π_2^* 是(1)(23)(4). 因此，

$$P_G(x_1,x_2)=\frac{1}{2}(x_1^4+x_1^2x_2).$$

令 $w(0)=a$，$w(1)=b$，我们发现有 $\sum_{r\in R}[w(r)]^k=a^k+b^k$. 因此，模式目录由下式给出：

$$\frac{1}{2}[(a+b)^4+(a+b)^2(a^2+b^2)]=a^4+3a^3b+4a^2b^2+3ab^3+b^4.$$

项 $3a^3b$ 表明存在三个指定 3 个 0 和一个 1 的开关函数模式. 读者也许希望确定这些模式.　484

8.6.4 波利亚定理的证明[⊖]

现在，我们给出波利亚定理的证明. 我们首先给出一系列引理. 本小节假设 $R=\{1, 2, \cdots, m\}$.

引理 8.2　*假设 D 被分割成互不相交的集合 D_1，D_2，…，D_p. 设 $C(D, R)$ 的子集 C 是由所有满足如下性质的着色 f 组成的：如果对于某个 i，a 和 b 都在 D_i 中，那么 $f(a)=f(b)$. 这时，集合 C 的目录由下式给出：*

$$\begin{aligned}&[w(1)^{|D_1|}+w(2)^{|D_1|}+\cdots+w(m)^{|D_1|}]\times[w(1)^{|D_2|}+w(2)^{|D_2|}+\cdots\\&+w(m)^{|D_2|}]\times\cdots\times[w(1)^{|D_p|}+w(2)^{|D_p|}+\cdots+w(m)^{|D_p|}].\end{aligned} \tag{8.24}$$

证明　把(8.24)式乘开，我们得到诸如下面的项：

$$w(i_1)^{|D_1|}w(i_2)^{|D_2|}\cdots w(i_p)^{|D_p|}.$$

这是给 D_1 的所有元素指定颜色 i_1，给 D_2 的所有元素指定颜色 i_2 等等的着色权. 因此，(8.24)式给出 D_i 的所有元素有相同颜色的着色权的和. ■

引理 8.3　*假设 $G^*=\{\pi_1^*, \pi_2^*, \cdots\}$ 是 $C(D, R)$ 的排列群. 对于 G^* 中的每一个 π^*，令 $\overline{w}(\pi^*)$ 是 $C(D, R)$ 中经过 π^* 保持不变的所有着色 f 的权的和. 假设 C_1，C_2，…是着色的等价类，而且 $w(C_i)$ 是 C_i 中所有 f 的共同权. 那么*

$$w(C_1)+w(C_2)+\cdots=\frac{1}{|G^*|}[\overline{w}(\pi_1^*)+\overline{w}(\pi_2^*)+\cdots]. \tag{8.25}$$

注意，如果所有的权都等于 1，那么引理 8.3 退化成伯恩赛德引理.

引理 8.3 的证明　(8.25)式的右边的和是对每一个 π^* 求经过 π^* 保持不变的所有着色 f 的权的和. 因此，在这里，$w(f)$ 被累加的次数正好是 f 经过某个 π^* 保持不变的次数. 使用 8.3.2 节的术语，这个数就是 f 的稳定算子中元素的数量 $\mathrm{St}(f)$. 根据 8.3.2 节的引理 8.1，$|\mathrm{St}(f)|=|G^*|/|C(f)|$，其中 $C(f)$ 是包含 f 的等价类. 因此，如果 $C(D, R)=\{f_1, f_2, \cdots\}$，那么

⊖⊖ 本小节可以跳过.

(8.25)式的右边由下式给出：

$$\frac{1}{|G^*|}[w(f_1)\cdot|\mathrm{St}(f_1)|+w(f_2)\cdot|\mathrm{St}(f_2)|+\cdots]=$$

$$\frac{1}{|G^*|}\left[w(f_1)\frac{|G^*|}{|C(f_1)|}+w(f_2)\frac{|G^*|}{|C(f_2)|}+\cdots\right],$$

这一表达式等于

$$\frac{w(f_1)}{|C(f_1)|}+\frac{w(f_2)}{|C(f_2)|}+\cdots. \tag{8.26}$$

485

如果对等价类 C_i 中的 f_i 求项 $w(f_i)/|C(f_i)|$ 的和，我们得到 $w(C_j)$，因为每一个 $w(f_i)=w(C_j)$，而且 $|C(f_i)|=|C_j|$．因此，(8.26)式等于 $w(C_1)+w(C_2)+\cdots$. ■

现在，我们来完成波利亚定理的证明. 在引理 8.3 的(8.25)式中，左边是模式目录. 回想一下，$\overline{w}(\pi^*)$是经过 π^* 保持不变的着色 f 的权的和. 假设排列 π 在其循环分解中有循环 D_1，D_2，…，D_p. 注意，着色 f 经过 π^* 保持不变，当且仅当，只要 a 和 b 是在相同的 D_i 中，则 $f(a)=f(b)$. 因此，根据引理 8.2，(8.24)式给出经过 π^* 保持不变的着色集合权的目录或和，即(8.24)式给出 $\overline{w}(\pi^*)$. (8.24)式中的每一项都有如下形式：

$$[w(1)]^j+[w(2)]^j+\cdots+[w(m)]^j=\sum_{r\in R}[w(r)]^j, \tag{8.27}$$

其中 $j=|D_i|$. 因此，项(8.27)式在(8.24)式中出现的次数与 $|D_i|$ 等于 j 的数目相同，即它与 π 有长度为 j 的循环的数目相同. 我们在 8.5.5 节中定义循环指数时，把这个数记为 b_j. 因此，$\overline{w}(\pi^*)$或(8.24)式可以重写成

$$\left[\sum_{r\in R}[w(r)]^1\right]^{b_1}\left[\sum_{r\in R}[w(r)]^2\right]^{b_2}\cdots.$$

因此，(8.25)式的右边变成

$$P_G\left(\sum_{r\in R}[w(r)]^1,\sum_{r\in R}[w(r)]^2,\cdots\right).$$

这就证明了波利亚定理.

本节练习

1. 如果 w(黑色)=3，w(白色)=4，确定图 8.1 的阵列 a 中每一个着色的权.
2. 如果 $w(1)=x$，$w(2)=y$，确定 8.4 节练习 6(a)和(b)中的着色 f 和 g 的权.
3. 假设 K 是由图 8.10 中的着色 C_2，C_8，C_{10} 和 C_{14} 组成的. 如果 w(黑)$=b$，w(白)$=w$，确定集合 K 的目录.
4. 假设 K 是由表 8.3 中的开关函数 T_2，T_3，T_8，T_{10}，T_{15} 组成的. 如果 $w(0)=a$，$w(1)=b$，确定 K 的目录.
5. 对于 8.1 节练习 9 的情况，如果 w(黑)$=b$，w(白)$=w$，确定模式目录.
6. 对于 8.4 节练习 7 的情况，如果 $w(1)=\alpha$，$w(2)=\beta$，确定模式目录.

486

7. 使用波利亚定理计算不同四个珠子项链的数量，其中每一个珠子有三种颜色中的一种颜色.
8. 对于例 8.4，假设对于顶点我们有四种可能的颜色. 使用波利亚定理确定这棵树的不同着色的数量.
9. 对于例 8.5，确定没有 CH_3 的不同分子的数量.
10. 有多少种满足下面条件的四个珠子项链：其中每一个珠子的颜色是 b，r 和 p 中的一种，而且至少有一个 p?
11. 考虑使用颜色 G，B 和 W 对 $K_{1,3}$ 的着色.
    (a)通过给出着色，验证不使用 G 的不同着色正好有 8 种.
    (b)通过给出着色，验证正好使用两个 W 的不同着色正好有 7 种.
    (c)确定正好使用一个 W 的不同着色的数量.

12. 确定8.2节练习23中介绍的下列各图的不同着色数量，假设我们使用颜色G，B和W而且只使用一个G.

    (a)L_4　　(b)Z_4　　(c)K_4-K_2

13. 使用颜色G，B，W，而且正好使用3个B确定练习12中各图的不同着色的数量.
14. 确定在值域内至少有一个1的不同二变量开关函数的数量，即这样的开关函数至少把1赋给一个位串.
15. 对于8.4节练习23，确定在值域内至少有一个1的不同三变量开关函数的数量.
16. 对于例8.5，确定至少有一个Cl原子且至少有一个H原子的不同分子的数量.
17. 使用波利亚定理计算满足下面条件的非同构图的数量：

    (a)三个顶点.

    (b)三个顶点和两条边.

    (c)三个顶点，至少有一条边.

    (d)四个顶点.

    (e)四个顶点和三条边.

    (f)四个顶点和至少两条边.

    (参见8.4节练习21和练习22.)关于波利亚理论对图论的进一步应用，参见Chartrand和Lesniak[1996]、Gross和Yellen[1999]、Harary[1969]、Harary和Palmer[1973].

18. 着色一个立方体的顶点，有5种颜色可用：红、蓝、绿、黄和紫. 计数至少使用一个绿色和一个紫色的不同着色的数量(参见8.5节练习22和23).
19. 设$D=\{a, b, c, d\}$，$R=\{0, 1\}$，且令G是由排列(1)(2)(3)(4)，(12)(34)，(13)(24)和(14)(23)组成的，并取$w(0)=1$，$w(1)=x$.

    (a)确定$C(D, R)$.

    (b)确定G^*.

    (c)确定G^*下的所有着色等价类.

    (d)确定G^*下的所有着色等价类的权.

    (e)设e_i是权x^i的颜色数量，而且$e(x)$是e_i的普通生成函数；即$e(x)=\sum_{i=0}^{\infty}e_ix^i$. 计算$e(x)$. [487]

    (f)设E_j是权x^j的模式的数量，而且$E(x)$是E_j的普通生成函数，即$E(x)=\sum_{j=0}^{\infty}E_jx^j$. 计算$E(x)$.

    (g)对于(e)和(f)中计算的$e(x)$和$E(x)$，证明：

$$E(x)=P_G[e(x),e(x^2),e(x^3),\cdots]. \tag{8.28}$$

20. 对于$D=\{a, b, c\}$，$R=\{0, 1\}$，及G是排列(1)(2)(3)和(12)(3)的集合，$w(0)=x^2$，$w(1)=x^7$，重复练习19.
21. 扩展练习19和20的结果，假设对于每一个r，存在非负整数p，使得$w(r)=r^p$. 设$e(x)$和$E(x)$如练习19中的定义. 证明(8.28)式成立.

参考文献

BOGART, K. P., *Introductory Combinatorics*, 3rd ed., Academic Press, San Diego, 1999.

BRUALDI, R. A., *Introductory Combinatorics*, 3rd ed., Prentice Hall, Upper Saddle River, NJ, 1999.

BURNSIDE, W., *Theory of Groups of Finite Order*, 2nd ed., Cambridge University Press, Cambridge, 1911. (Reprinted by Dover, New York, 1955.)

CAMERON, P. J., "Automorphism Groups of Graphs," in L. W. Beineke and R. J. Wilson (eds.), *Selected Topics in Graph Theory, 2*, Academic Press, London, 1983, 89–127.

CHARTRAND, G., and LESNIAK, L., *Graphs and Digraphs*, 3rd ed., CRC Press, Boca Raton, 1996.

DE BRUIJN, N. G., "Generalization of Pólya's Fundamental Theorem in Enumerative Combinatorial Analysis," *Ned. Akad. Wet., Proc. Ser. A 62, Indag. Math., 21* (1959), 59–79.

DE BRUIJN, N. G., "Pólya's Theory of Counting," in E. F. Beckenbach (ed.), *Applied Combinatorial Mathematics*, Wiley, New York, 1964, 144–184.

GROSS, J., and YELLEN, J., *Graph Theory and Its Applications*, CRC Press, Boca Raton, FL, 1999.

HARARY, F., *Graph Theory*, Addison-Wesley, Reading, MA, 1969.

HARARY, F., and PALMER, E. M., *Graphical Enumeration*, Academic Press, NY, 1973.

HARRISON, M. A., *Introduction to Switching and Automata Theory*, McGraw-Hill, New York, 1965.

LIU, C. L., *Introduction to Combinatorial Mathematics*, McGraw-Hill, New York, 1968.

PÓLYA, G., "Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und chemische Verbindungen," *Acta Math., 68* (1937), 145–254.

PÓLYA, G., and READ, R. C., *Combinatorial Enumeration of Groups, Graphs, and Chemical Compounds*, Springer-Verlag, New York, 1987.

PRATHER, R. E., *Discrete Mathematical Structures for Computer Science*, Houghton Mifflin, Boston, 1976.

REDFIELD, J. H., "The Theory of Group-Reduced Distributions," *Amer. J. Math., 49* (1927), 433–455.

REINGOLD, E. M., NIEVERGELT, J., and DEO, N., *Combinatorial Algorithms: Theory and Practice*, Prentice Hall, Englewood Cliffs, NJ, 1977.

STONE, H. S., *Discrete Mathematical Structures and Their Applications*, Science Research Associates, Chicago, 1973.

488

TUCKER, A. C., *Applied Combinatorics*, 3rd ed., Wiley, New York, 1995.

第三部分　存在问题

第9章　组合设计

9.1　区组设计

在尝试进行科学、合理的实验的历史中，组合数学起到了重要的作用．我们已经在1.1节遇到过实验设计问题，在那里我们讨论了药物效应研究中的实验设计问题，并使用这一设计问题引入了拉丁方的概念．本章我们研究在实验设计中产生的组合问题，并讨论实验设计理论中组合分析的作用．第10章将把组合设计运用于错误校正码理论中．本章还将遇到的其他应用有：密码学、计算机和网络通信、软件测试、磁盘阵列存储、信号处理、运动会进程表、不合格产品的“组群测试”，DNA探测芯片的设计以及分子生物学中的无性筛选．其中的一些应用在Stinson[2003]、Colbourn，Dinitz和Stinson[1999]中有所描述．关于组合设计的一般参考文献，请参见Anderson[1990]、Beth，Jungnickel和Lenz[1999]、Colbourn和Dinitz[1996]、Dinitz和Stinson[1992]、Hughes和Piper[1988]、Lindner和Rodger[1997]、Street和Street[1987]、Wallis[1988]．

实验设计理论很大程度上产生于R. A. Fisher和F. Yates以及其他人的工作，他们的工作受到了农业中的农田实验设计问题的启发．尽管目前这一理论的适用性已经相当广泛，但是其中的很多术语仍然留有其原本的印迹．

我们将集中研究以比较不同*处方*(treatment)或*变元*(variety)的影响为目的的实验，例如，不同类型的肥料，一种药物的不同剂量或不同品牌的鞋或轮胎等．将每一处方运用于若干*实验单元*(experimental unit)或*区域*(plot)中．在农业方面，实验单元可能是一块长着庄稼的土地．然而，实验单元也可能是在某一天的某个人类对象、一片动物组织或者是接受注射或化学处理的动物身体或植物体上的部位，在其他情况下实验单元也可能是为某种目的在某个位置使用的机器．

某些实验单元被组合到一起形成*区组*(block)．通常，把这些单元组合成区组是因为它们有某些共同的固有性质：例如，它们都是在同一人类对象上，或者都是在农田中同一垄上，或者它们都是在相同动物的皮肤上，或都是在相同的机器上．

为了更具体些，我们考虑比较四种不同轮胎品牌的胎面磨损问题．[⊖]我们要比较的处方是四种不同品牌的轮胎．显然，给定品牌的轮胎也许存在个体差．因此，我们当然希望对每一个品牌的多个轮胎进行实验．特定的轮胎是一个实验单元．现在，假设我们在实际驾驶条件下进行测试．于是，我们自然地把四个轮胎(即实验单元)组合到一起，因为进行轮胎测试的汽车(通常)使用四个轮胎．测试汽车定义为区组．

很自然地，我们设法让每个品牌的轮胎(即处方)使用的次数一样多．假设每一种轮胎的使用次数为r．那么我们总共需要$4r$个实验单元，因为有四种轮胎品牌．因为实验单元被分成大小

⊖　我们采用Hicks[1973]的处方．

为 4 的区组，所以 $4r$ 一定被 4 整除．对于本例的情况，r 可以是任意的正整数．如果有五个轮胎品牌，那么我们总共需要 $5r$ 个实验单元，因此，我们只能选取 r 是使得 $5r$ 能被 4 整除的整数．

如果我们取 r 为 4，那么我们就可以有一个非常简单的实验设计．设四辆汽车为 A，B，C，D，而且把品牌 1 的四个轮胎安装到汽车 A 上，把品牌 2 的四个轮胎安装到汽车 B 上，把品牌 3 的四个轮胎安装到汽车 C 上，把品牌 4 的四个轮胎安装到汽车 D 上．这一设计概括在表 9.1 中．显然，这不是一个令人满意的实验设计．不同的汽车(和不同的驾驶员)可能导致不同的轮胎磨损量，这个实验设计将受到外界因素的困扰．

表 9.1　测试轮胎磨损的实验设计

轮胎位置	汽车	A	B	C	D
	左前	1	2	3	4
	右前	1	2	3	4
	左后	1	2	3	4
	右后	1	2	3	4

注：i，j 项是第 j 辆汽车的位置 i 上使用的轮胎的品牌．

[490] 实验设计理论的许多目标是消除由于特定实验单元中的变更所引起的偏差或混乱效应．人们一般通过随机化或以随机的方式对实验单元指定实施处方来消除这样的效应．例如，我们可以用完全随机的方式给每一辆汽车指定轮胎．这可能导致如表 9.2所示的设计．可是，正如表 9.2 所示，我们的实验中某种品牌(例如品牌 4)从没有在特定的汽车(例如汽车 A 上)使用；或者某个品牌，例如品牌 3，在诸如汽车 A 中被使用了多次．这样的结果仍然有偏差．如果我们要求每一种品牌必须被用于每个汽车上，然后，对汽车轮子随机地分配轮胎，那么我们就能够避免这样的情况．实验设计理论的一个主要问题就是我们在第 1 章所说的存在问题．本章我们以如下形式提出这个问题：是否存在一个这样的设计，在这一设计中有四个品牌和四辆汽车，每一品牌使用 4 次，而且在每一辆汽车上它至少被使用一次，同时正好使用一次？答案是有这样的设计存在．表 9.3 给出一个这样的设计．

表 9.2　测试轮胎磨损的随机设计

轮胎位置	汽车	A	B	C	D
	左前	3	4	2	2
	右前	1	1	4	4
	左后	3	4	1	3
	右后	2	3	2	1

注：i，j 项是用在汽车 j 的 i 位置上的轮胎品牌．

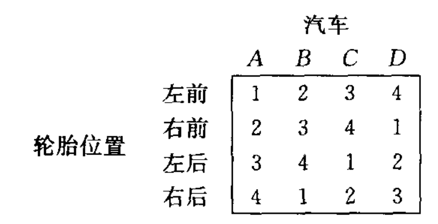
表 9.3　测试轮胎磨损的完全区组设计

轮胎位置	汽车	A	B	C	D
	左前	1	1	3	4
	右前	2	3	4	2
	左后	3	2	1	1
	右后	4	4	2	3

注：i，j 项是用在汽车 j 的 i 位置上的轮胎品牌．

表 9.3 的设计仍然有一些缺点．轮胎在汽车上的位置会影响这一轮胎的寿命．例如，后胎与前胎受到的磨损不同，甚至轮胎在汽车的哪一侧也会影响它的寿命．如果我们还希望消除轮子 [491] 位置的偏差效应，可以要求每一品牌正好在每一辆汽车上使用一次，而且还在每一个可能位置上正好使用一次．这时，我们要求把数字 1，2，3，4 分配到 4×4 数组中，且每一个数字正好在数组的每一行和每一列中出现一次．即我们要求有一个拉丁方(参见 1.1 节)．表 9.4 给出一个这样的设计．在所有可能的 4×4 拉丁方设计中，我们仍然可以随机地挑选出一个特定的拉丁方来使用．

表 9.4　测试轮胎磨损的拉丁方设计



轮胎位置	汽车	A	B	C	D
	左前	1	2	3	4
	右前	2	3	4	1
	左后	3	4	1	2
	右后	4	1	2	3

注：i，j 项是用在汽车 j 的 i 位置上的轮胎品牌．

在某些实验中，不可能把所有处方运用于每一个区组上．例如，如果有 5 个品牌的轮胎，那么我们只能在每个区组中只使用其中的四种．现在，我们如何设

计一个实验呢？如果每一个品牌的轮胎被使用 r 次，那么我们要把总共 $5r$ 个轮胎分成四组，正如我们上面所观察的那样，$5r$ 必须能被 4 整除. 例如，r 必须等于 4，8，12 等. 注意，我们不能使用 6 辆汽车来做实验，即不存在这样的实验设计：使用 5 个品牌和 6 辆汽车，而且每个品牌使用的次数相同，且 4 个(不同)品牌被分配到每一辆汽车上. 因为总共存在 24 个轮胎位置，而 $5r=24$ 是不可能的. 假设 $r=4$. 那么共有 $5r=20$ 个轮胎位置. 假设 s 是汽车的数量，那么 $4s=20$，所以 s 等于 5. 表 9.5 给出一个可能的设计. 这里，每一辆汽车上有四个不同品牌的轮胎，每一个品牌在每一个位置上正好使用一次，每一个品牌的使用次数相同，都等于 5. 我们可以对这样的设计提出各种额外的要求. 下面我们讨论其中的一些要求.

表 9.5　测试轮胎磨损的不完全区组设计

		汽车				
		A	B	C	D	E
轮胎位置	左前	1	2	3	4	5
	右前	2	3	4	5	1
	左后	3	4	5	1	2
	右后	4	5	1	2	3

注：i，j 项是用在汽车 j 的 i 位置上的轮胎品牌.

现在我们介绍一些通用术语. 假设 P 是实验单元或区域的集合，V 是处方或变元的集合. P 的特定子集称为区组(block). 给定 P 和 V，一个区组设计(block design)定义为给出一系列区组，并把 V 中的处方分配给 P 中的每一个实验单元. 因此，相应的每一个区组是一个(有可能有重复)处方的集合. 抽象地说，我们可以忽视实验单元，并把一个区组设计简单地考虑为由处方的集合 V 和一系列称为区组的 V(可能有重复)的子集组成的. 因此，对应于表 9.2 的区组设计有 $V=\{1, 2, 3, 4\}$，且有下面的区组：

$$\{3, 1, 3, 2\}, \{4, 1, 4, 3\}, \{2, 4, 1, 2\}, \{2, 4, 3, 1\}.$$

492

如果考虑顺序，如在拉丁方中那样，那么我们可以把这些区组考虑成序列而不是子集. 区组设计称为是完全的(complete)，如果每一个区组是所有的 V，否则称为不完全的(incomplete). 表 9.3 和表 9.4 定义完全区组设计，而表 9.5 则定义一个不完全区组设计. 一个区组设计称为是随机化的(randomized)，如果每一个区组中的元素是由某个随机装置排序的，诸如随机数表或设计用于选取随机排列的计算机程序等.

本章研究两种类型的区组设计，即来自于拉丁方和拉丁方族的完全设计和平衡不完全设计. 我们还把实验设计与被称为有限投影平面的有限几何的研究联系起来. 第 10 章将把有关实验设计的结果运用于错误校正码的设计中.

本节练习

1. 确定一个不同于表 9.4 给出的轮胎磨损实验的拉丁方设计.
2. 假设我们希望测试 6 种不同过敏性药物的效应. 每一个对象在一周的每一天领到一种药.
   (a)变元、实验单元和区组各是什么？
   (b)如果一个实验把每一种药物分配给相同数量的对象，那么所需对象的数量是多少？
   (c)这一实验有拉丁方设计吗？
   (d)有在一周内每一种药物的使用次数相同的实验设计吗？如果有，区组是什么？
   (e)有每一种药物在相同数量的对象中使用的实验设计吗？
   (f)这一实验有每个对象得到每一种药物的次数相同的设计吗？

493

3. (a)(作为子集)给出变元为{1，2，3，4}且区组为 3 子集的设计的区组.
   (b)重复(a)使得每一个变元恰好出现在 3 个区组中.
4. (作为子集)给出变元为{1，2，3，4，5，6，7，8，9}，区组为 3 子集，且每一个变元正好出现在 4 个区组中的设计的区组.
5. (a)是否存在有 8 个变元，区组的大小为 4，而且每个变元只在一个区组中出现的设计？如果存在这样的

设计，那么(作为子集)给出这一设计的区组. 否则，解释为什么不存在这样的设计.

(b)如果有 12 个变元，区组的大小为 5，且每个变元只出现在 1 个区组中，重复(a).

(c)如果有 7 个变元，区组的大小为 3，且每个变元正好出现在 3 个区组中，重复(a).

(d)如果有 10 个变元，区组的大小为 4，且每个变元正好出现在 2 个区组中，重复(a).

6. (a)对于变元为{1, 2, 3, 4, 5, 6, 7, 8}，且区组为所有 3 子集的设计，每个变元在多少个不同区组中出现?

(b)一般地，如果有 v 个变元，且区组为所有 k 子集，那么每个变元在多少个区组中出现?

7. 假设有一种区组设计，在这个区组设计中 v 个变元中的每一个出现在 r 个区组中，且区组是 k 子集. 考虑一个与此相关联的区组设计，在这个区组设计中，区组是原来区组的补. 使用变元和区组的数量、每一个区组的大小以及每一个变元出现的区组的数量来描述这一新区组设计.

9.2 拉丁方

9.2.1 一些例子

如果存在两个因素，例如，对象和日期、轮的位置和汽车或者行(row)和列(column)，而且我们要对这两个因素进行控制，那么拉丁方设计是合适的实验设计. 在农业实验中，行和列就是矩形农田的行和列. Fisher[1926]引入拉丁方研究这样的实验. 例如，假设存在 k 种不同的行效应和 k 种不同的列效应，而我们希望测试 k 种不同的处方. 我们希望安排这样的事件，使得每一种处方在给定的行和列出现且仅出现一次，例如，在一台给定的汽车和给定的位置上. 显然，对每一个 k，存在这样的安排，或 $k\times k$ 拉丁方. 表 9.6 给出一个 $k\times k$ 拉丁方. 因此，对于拉丁方，其存在问题非常简单. 而对本章所考虑的其他设计，存在问题并不是那样简单.

表 9.6 $k\times k$ 拉丁方

1	2	3	⋯	$k-1$	k
2	3	4	⋯	k	1
3	4	5	⋯	1	2
			⋮		
$k-1$	k	1	⋯	$k-3$	$k-2$
k	1	2	⋯	$k-2$	$k-1$

494 现在，我们给出一系列使用拉丁方设计的例子.

【例 9.1 假牙修复术】 Cox[1958]⊖讨论了比较 7 种处方的假牙修复术的一项实验，其中的处方是不同材料的商业假牙及其安装的不同角度. 这一实验希望尽可能地消除由于患者之间的差异而导致的偏差. 因此，每一位患者试带一种假牙一个月，然后再试带另一种假牙一个月，以此类推. 7 个月以后，每一位患者都已试带过每一种类型的假牙，即是每一个处方的对象.

在这个实验中，看起来后几个月的结果不同于前几个月的结果，因此，应该在每一个时间段均等地使用每一个处方. 因此，有两种类型的偏差：患者之间的偏差和时间之间的偏差. 要抵消这两个类型偏差建议使用拉丁方. 这里的行对应于月份，而列对应于患者. 每一位患者定义一个区组，而实验单元是在第 i 个月中的第 j 个患者. ■

【例 9.2 心脏药品】 Chen，Bliss 和 Robbins[1942]在猫身上测试了 12 种不同心脏药品的效果. 这一实验要求一个观察者仔细测量一段时间内的药效，所以指定一名观察者一天只能观察 4 只不同的猫. 实验者希望消除观察日期的效应、观察者的效应，以及观察日期的时间段(早晨、上午、下午、傍晚)的效应. 因此，这里有三个因素，这不适合于拉丁方设计. 然而，可以取观察日期为第一个因素，而以观察员和观察当天的具体时间作为第二个因素来实现拉丁方设计. 进行 12×12 拉丁方实验，历时 12 天，3 个观察员中每一位观察员每一天观察 4 只猫，上午两只下午两只. 这一设计有 12 行，以观察员和观察时间编码，有 12 列，以日期编码. i，j 项是观察时段的日期 j 及编码为 i 的观察员观察的药物. 该日期定义区组. ■

495

⊖ 下面的例 9.2、例 9.3、例 9.5、例 9.13 和例 9.14 也是由 Cox[1958]讨论的. 这些例子和其他例子还可以在 Box，Hunter 和 Hunter[1978]、Cochran 和 Cox[1957]、Finney[1960]、Hicks[1973]中找到.

【例 9.3 市场调研】 Brunk 和 Federer[1953]讨论了对市场调研的一些实验. 其中之一是研究改变价格、摆放以及包装的操作对苹果销售的影响. 在所进行的一系列实验的每一个实验中，要对四种推销方法(处方)1，2，3 和 4 进行比较，且有 4 个市场参与. 显然，我们期望每一个商店使用每一种处方，所以应该安排这一实验持续时间是四个时间周期的倍数. 实验者希望消除商店之间和周期之间的系统差异. 因为存在两种类型的偏差，所以拉丁方设计，特别是 4×4 拉丁方设计是适当的. 然而，事实上，他们把一周分成两个部分，星期一到星期四为第一部分，星期五到星期六为第二部分，然后，为实验周的每一部分建立一个 4×4 拉丁方. 这一想法很好，因为每一位消费者在周末的食品订购会更大，而且也完全有可能在实验的两个部分中处方的差异不相同. 对于一项持续一周并进行四个处方比较的实验，他们使用了表 9.7 的设计.

表 9.7 市场调研实验中一周两个部分的两个拉丁方设计

一周中的前半部分

商店	星期一	星期二	星期三	星期四
A	2	1	4	3
B	3	2	1	4
C	4	3	2	1
D	1	4	3	2

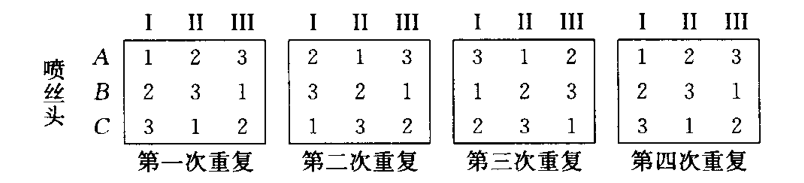
一周中的后半部分

商店	星期五上午	星期五下午	星期六上午	星期六下午
A	2	3	1	4
B	1	4	2	3
C	3	2	4	1
D	4	1	3	2

注：i，j 项给出在第 j 阶段的第 i 家商店使用的处方. ■

【例 9.4 纺织人造丝】 Box，Hunter 和 Hunter[1978]讨论了一项关于人造丝破损强度以及改变拉伸比，即人造丝纺织时施加的张力，是如何影响破损度的实验. 要测试的三个处方是(1)普通的拉伸比，(2)拉伸比提高 5%，(3)拉伸比提高 10%. 使用一种纺织机，该机有三个不同的喷丝头在三种不同的拉伸比下为三个不同的线轴供丝. 当所有线轴都完全绕满丝时，将替换上新的空线轴，并继续实验. 实验员希望控制两个因素：三个不同喷丝头的效应，以及喷丝头使用的时间(次序)的效应. 这需要 3×3 拉丁方设计，其中列的标签为 I、II、III，对应于生产丝的次序，其行的标签为 A、B、C，对应于所使用的喷丝头. i，j 项是第 j 次生产时第 i 个喷丝头在产丝时所使用的拉伸比(1，2 或 3)，或处方. 当使用小拉丁方时，通常希望重复它们，所以，事实上他们的实验重复了四次，使用不同的 3×3 拉丁方设计. 他们使用了表 9.8 所示的设计. 496

表 9.8 人造丝实验的拉丁方设计



生产的次序

喷丝头	第一次重复 I	II	III	第二次重复 I	II	III	第三次重复 I	II	III	第四次重复 I	II	III
A	1	2	3	2	1	3	3	1	2	1	2	3
B	2	3	1	3	2	1	1	2	3	2	3	1
C	3	1	2	1	3	2	2	3	1	3	1	2

注：每一个拉丁方的 i，j 项是在第 j 次生产时第 i 个喷丝头使用的拉伸比. ■

9.2.2 正交拉丁方

我们再一次返回到在 9.1 节所讨论的四种品牌的轮胎磨损例子. 想象一下，我们还对制动衬面对轮胎磨损的效应感兴趣. 为简便起见，假设我们还有四种不同品牌的制动衬面. 因此，我们除了希望在每一辆汽车及每一个位置对每一个品牌的轮胎正好测试一次之外，还希望对每一种品牌的轮胎与每一种品牌的制动衬面的组合正好测试一次. 我们可以借助于构建一个 4×4 数组来完成上面的任务，其中数组的行对应于轮子的位置，而列对应于汽车，而且在这个数组的每一

个格子处放置一种品牌的轮胎和一种品牌的制动衬面，它们被用于相应的汽车和相应的位置上. 假设 a_{ij} 是这个数组的 i，j 项处的轮胎品牌，而且 b_{ij} 是这一项中的制动衬面品牌. 如果我们列出所有序对(a_{ij}，b_{ij})，我们要求每一个可能的轮胎品牌 a 和制动衬面品牌 b 的序对(a，b)都出现. 等价地，因为有 $4\times4=16$ 种可能的序对(a，b)，而且在这一数组中正好有 16 个位置，所以我们要求所有序对(a_{ij}，b_{ij})互不相同. 我们能做到这一点吗? 我们肯定能够做到这一点. 如果制动衬面的品牌记为 1，2，3，4，那么可以简单地在第 i 辆汽车上的每一个轮子上测试制动衬面品牌 i. 把这一设计与表 9.4 给出的轮胎品牌设计结合起来，给出表 9.9 的序对数组. 该表中的所有序对都是互不相同的.

表 9.9 轮胎品牌和制动衬面品牌对胎面磨损的综合效应的测试设计

		汽车			
		A	B	C	D
轮胎位置	左前	(1, 1)	(2, 2)	(3, 3)	(4, 4)
	右前	(2, 1)	(3, 2)	(4, 3)	(1, 4)
	左后	(3, 1)	(4, 2)	(1, 3)	(2, 4)
	右后	(4, 1)	(1, 2)	(2, 3)	(3, 4)

注：i、j 项是一个序对，第一个元素是在第 j 辆汽车的第 i 个位置上所使用的轮胎的品牌，第二个元素是在那里所使用的制动衬面的品牌.

可是，如果我们只考虑制动衬面的话，表 9.9 的数组不是一个十分令人满意的设计. 因为我们只在汽车 A 上使用品牌 1 的制动衬面，在汽车 B 上使用品牌 2 的制动衬面等. 如果通过拉丁方设计进行制动衬面测试，而不只进行轮胎测试，那么这个设计就完美了. 因此，我们希望寻找两个拉丁方实验 $A=(a_{ij})$和 $B=(b_{ij})$，其中一个是轮胎品牌测试的拉丁方，而另一个是制动衬面品牌测试的拉丁方，这两个拉丁方设计都使用相同的行和列效果. 另外，我们希望这些序对(a_{ij}，b_{ij})各不相同. 能够做到这一点吗? 在我们的例子中，这一点是能够做到的. 表 9.10 给出两个拉丁方设计和相应序对组成的数组，很容易看到，每一个序对(a，b)，有 $1\leqslant a\leqslant4$，$1\leqslant b\leqslant 4$，正好出现一次. 等价地，这些序对各不相同. 我们称两个不同的 $n\times n$ 拉丁方 $A=(a_{ij})$和 $B=(b_{ij})$是*正交的*(orthogonal)，如果 n^2 个序对(a_{ij})各不相同. 因此，表 9.10 中的两个 4×4 拉丁方是正交的. 然而，表 9.7 的两个拉丁方不是正交的，因为序对(2，4)出现两次，一次在位置 2，2

表 9.10 测试轮胎品牌和制动衬面品牌对轮胎磨损的综合效应的两个正交拉丁方设计

		汽车			
		A	B	C	D
轮胎位置	左前	1	2	3	4
	右前	2	1	4	3
	左后	3	4	1	2
	右后	4	3	2	1

	汽车			
	A	B	C	D
左前	4	1	2	3
右前	3	2	1	4
左后	1	4	3	2
右后	2	3	4	1

		汽车			
		A	B	C	D
轮胎位置	左前	(1, 4)	(2, 1)	(3, 2)	(4, 3)
	右前	(2, 3)	(1, 2)	(4, 1)	(3, 4)
	左后	(3, 1)	(4, 4)	(1, 3)	(2, 2)
	右后	(4, 2)	(3, 3)	(2, 4)	(1, 1)

综合设计

注：组合数组在 i，j 项处的序对的第一个元素是汽车 j 的 i 位置处使用的轮胎品牌，第二个元素是汽车 j 的 i 位置处使用的制动衬面品牌，它们得自前两个拉丁方.

上，一次在位置3，3上. 更一般地，如果$A^{(1)}$，$A^{(2)}$，…，$A^{(r)}$是不同的$n\times n$拉丁方，则称它们形成一个*正交族*(orthogonal family)，如果其中的每一对拉丁方都是正交的.

本节讨论的主要问题是基本的存在问题：如果我们想要使用一对正交的$n\times n$拉丁方来设计一项实验，我们总能肯定存在这样的一对拉丁方吗？更一般地，我们要问：什么时候存在r个不同的$n\times n$拉丁方的正交族？

在阐述这些问题之前，我们给出几个使用正交拉丁方设计的例子.

【例9.5 油耗】 Davies[1945]使用一对正交拉丁方，对于不同等级的汽油的每加仑里程的油耗进行了比较. 一共对7个等级的汽油进行了测试. 整个过程使用一辆汽车. 每一次测试驾驶测试汽车在固定道路上跑20英里，路程包括不同的坡道. 为了消除因驾驶员而带来的偏差，使用了7名驾驶员；而为了消除交通条件所带来的效应，在不同日期的七个不同的时间段进行实验. 因此，在7个比较的处方之外，还存在三个实验单元分类：驾驶员分类、日期分类和测试日的时间段分类. 双重实验单元分类表明要使用拉丁方，而三重实验单元分类表明要使用一对正交拉丁方. 后者考虑这样的实验，在这个实验中，每一个等级的汽油每天使用一次，被每一名驾驶员使用一次，而且在测试日的每个时间段使用一次，从而保证进行均衡的比较. 这一设计给每一天(行)和每一个时间段(列)(在第一个拉丁方里)分配一个汽油等级和(在第二个拉丁方里)一名司机(在9.1节的轮胎磨损例子中，我们不能用相同的方法控制司机，即使用如下的正交拉丁方实验是没有意义的：第一个拉丁方表示汽车j的位置i的轮胎品牌，第二拉丁方表示汽车j的位置i的司机. 因为同一名司机必定指定给给定汽车的所有位置). ■

【例9.6 布料测试】 Box，Hunter和Hunter[1978]描述了关于Martindale磨损测试机的实验，这是用于测试诸如布料等材料的磨损量的机器. 在考虑类型的Martindale磨损测试机的一次运行中，机器同时用砂纸摩擦四块布料，然后测量所失去的重量. 有4个不同的测试台，标为A，B，C，D，而且每一个测试台可以用于测试机的四个位置P_1，P_2，P_3，P_4中的一个. 在一次实验中，要比较4种布料或处方，标记为1，2，3，4. 实验员希望控制4个不同的样品测试台、机器的4个位置、布料被测试的次序以及摩擦布料的砂纸. 四重实验单元分类表明要使用3个正交的4×4拉丁方族. 决定使用4张砂纸，标记为α，β，γ，δ，并把每一张砂纸切成4等块，在一个实验单元中使用其中的一块. 测试机运行4次，记为R_1，R_2，R_3，R_4，每一次运行在不同位置上使用不同的测试台和不同的1/4块砂纸块来测试4种布料样本. 表9.11给出所使用的3个拉丁方设计. 读者可以验证这些拉丁方都是两两正交的(事实上，实验被重复进行，在另一组3个正交4×4拉丁方族的设计下，另外使用4次以上的测试机和4张以上的砂纸).

497～499

表9.11 测试布料磨损的三个正交拉丁方族

处方的拉丁方设计（次序）

位置	R_1	R_2	R_3	R_4
P_1	1	3	4	2
P_2	2	4	3	1
P_3	3	1	2	4
P_4	4	2	1	3

测试台的拉丁方设计

位置	R_1	R_2	R_3	R_4
P_1	A	D	B	C
P_2	B	C	A	D
P_3	C	B	D	A
P_4	D	A	C	B

砂纸的拉丁方设计

位置	R_1	R_2	R_3	R_4
P_1	α	β	γ	δ
P_2	β	α	δ	γ
P_3	γ	δ	α	β
P_4	δ	γ	β	α

注：i，j项分别表示在位置P_i的运行R_j中使用的处方(布料类型)、测试台和砂纸. ■

9.2.3　有关正交族的存在结果

设一个 $n\times n$ 的拉丁方的秩(order)是 n. 在下面的讨论中，我们通常假设秩为 n 的拉丁方中的项是整数 1，2，…，n. 现在，我们讨论这样的问题：是否存在 r 个秩为 n 的拉丁方正交族？我们假设 $n>1$，因为只存一个 1×1 拉丁方. 不存在正交的 2×2 拉丁方对. 唯一的秩为 2 的拉丁方如表 9.12 所示. 它们不是正交的，因为序对(1，2)出现两次. 我们已在表 9.10 中看到，存在秩为 4 的正交拉丁方对，而在表 9.11 中我们看到存在秩为 4 的三个拉丁方的正交族. 很容易给出秩为 3 的正交拉丁方对(请读者尝试).

表 9.12　秩为 2 的两个拉丁方

1	2
2	1

2	1
1	2

第一个定理给出秩为 n 时存在 r 个拉丁方的正交族的必要条件.

定理 9.1　*如果存在秩为 n 的 r 个拉丁方的正交族，那么 $r\leqslant n-1$.*

证明　假设 $A^{(1)}$，$A^{(2)}$，…，$A^{(r)}$ 形成 $n\times n$ 拉丁方的正交族. 设 $a_{ij}^{(p)}$ 是 $A^{(p)}$ 的 i，j 项. 重新标记第一个拉丁方中的各项，使得 1 就在 1，1 处，即 $a_{11}^{(1)}=1$. 按如下方法完成这一工作. 如果 $a_{11}^{(1)}$ 等于 k，那么在整个 $A^{(1)}$ 中把 1 与 k 交换，且 k 与 1 交换. 这样做的结果仍是拉丁方，而且也不改变正交性，因为如果序对

500

$$(a_{ij}^{(1)},a_{ij}^{(p)})$$

是(k，l)，那么经转换后它变成(1，l)，而如果它是(1，l)，则现在它变成(k，l).

根据同样的推理，在保持 r 个 $n\times n$ 拉丁方正交的前提下，我们能够做适当的安排使得在每一个拉丁方的每一个 1，1 项都是 1，更一般地，使得

$$a_{11}^{(1)}=a_{11}^{(2)}=\cdots=a_{11}^{(r)}=1,$$
$$a_{12}^{(1)}=a_{12}^{(2)}=\cdots=a_{12}^{(r)}=2,$$
$$a_{13}^{(1)}=a_{13}^{(2)}=\cdots=a_{13}^{(r)}=3,$$
$$\vdots$$
$$a_{1n}^{(1)}=a_{1n}^{(2)}=\cdots=a_{1n}^{(r)}=n.$$

即我们能够做适当的安排使得每一个 $A^{(p)}$ 都有相同的第一行：

$$1\ 2\ 3\cdots n.$$

让我们考虑每一个拉丁方中的 2，1 项. 因为 $A^{(p)}$ 是拉丁方，而且因为 $a_{11}^{(p)}$ 是 1，而 1 在一列中只能出现一次，所以 $a_{21}^{(p)}$ 一定不等于 1. 另外，根据正交性，如果 $p\neq q$，有

$$a_{21}^{(p)}\neq a_{21}^{(q)},$$

否则，对于某个 i，有

$$(a_{21}^{(p)},a_{21}^{(q)})=(i,i),$$

所以

$$(a_{21}^{(p)},a_{21}^{(q)})=(a_{1i}^{(p)},a_{1i}^{(q)}),$$

这违反正交性. 因此，数

$$a_{21}^{(1)},a_{21}^{(2)},\cdots,a_{21}^{(r)}$$

各不相同，且都不等于 1. 因此至多存在 $n-1$ 个这样的数，且 $r\leqslant n-1$. (形式上，这一推导使用了 2.19 节的鸽巢原理.)　■

501

我们从表 9.13 所示的秩为 4 的 3 个拉丁方的正交族开始来说明这一定理的证明. 该表说明了排列所有的第一行都为 1234 的过程. 注意表 9.13 的最后一行的 3 个拉丁方的 2，1 项分别是 2，3 和 4.

表 9.13 把拉丁方正交族转换成每一个拉丁方在其第一行都为 123…n 的拉丁方正交族的过程

$A^{(1)}$				$A^{(2)}$				$A^{(3)}$			
4	3	2	1	2	1	4	3	3	4	1	2
3	4	1	2	4	3	2	1	2	1	4	3
2	1	4	3	3	4	1	2	4	3	2	1
1	2	3	4	1	2	3	4	1	2	3	4

↓ 1和4交换 ↓ 　　↓ 1和2交换 ↓ 　　↓ 1和3交换 ↓

1	3	2	4	1	2	4	3	1	4	3	2
3	1	4	2	4	3	1	2	2	3	4	1
2	4	1	3	3	4	2	1	4	1	2	3
4	2	3	1	2	1	3	4	3	2	1	4

↓ 2和3交换 ↓ 　　↓ 3和4交换 ↓ 　　↓ 2和4交换 ↓

1	2	3	4	1	2	3	4	1	2	3	4
2	1	4	3	3	4	1	2	4	3	2	1
3	4	1	2	4	3	2	1	2	1	4	3
4	3	2	1	2	1	4	3	3	4	1	2

定理 9.1 说明我们不能找到由多于 $n-1$ 个拉丁方组成的 $n\times n$ 拉丁方的正交族．我们说秩为 n 的拉丁方的正交族是*完全的*(complete)，如果这个正交族有 $n-1$ 个拉丁方．因此，表 9.13 所示的秩为 4 的 3 个拉丁方组成一个完全正交族．通常认为一个 2×2 拉丁方就构成一个正交族．

定理 9.2 给出拉丁方完全正交族存在的充分条件．

定理 9.2 *如果 $n>1$ 且 $n=p^k$，其中 p 是素数*⊖*，而 k 是一个正整数，那么存在秩为 n 的拉丁方的完全正交族．*

在此，我们跳过定理 9.2 的证明．我们在 9.3.4 节和 9.3.5 节给出此定理的证明，在那里，我们通过给出当 n 是素数的幂时，确定秩为 n 的拉丁方完全正交族的构造过程来证明这一定理．特别地，定理 9.2 表明存在一对正交 3×3 拉丁方，因为 $3=3^1$．它还说明存在 3 个两两正交的 4×4 拉丁方，因为 $4=2^2$(我们已在表 9.11 和表 9.13 看到了 3 个这样的拉丁方)．还存在 4 个两两正交的 5×5 拉丁方族，因为 $5=5^1$．因为 6 不是素数的幂，定理 9.2 不能告诉我们是否存在 5 个两两正交的 6×6 拉丁方的集合，甚至是否确实存在一对这样的拉丁方．下面将证明如果 $n=6$，那么不存在秩为 n 的拉丁方完全正交族，实际上甚至不存在一对这样的拉丁方．因此，对于 $n\leqslant 9$，当且仅当 $n\neq 6$ 时，存在秩为 n 的拉丁方完全正交族．Lam，Thiel 和 Swiercz[1989]通过大量的计算机搜索，确定不存在秩为 10 的拉丁方完全正交族．如果 $n=11$，则存在拉丁方完全正交族．因此，这就使得 $n=12$ 成为我们不知道是否存在完全正交族的最小数．在写这本书时，我们知道的最好结果是可能存在 5 个秩为 12 的两两正交的拉丁方(参见 Jonhson，Dulmage 和 Mendelsohn[1961])．

502

根据代数的基本定理，任何大于 1 的整数 n 都可以唯一地写成素数的(整数)幂的积：

$$p_1^{t_1} p_2^{t_2} \cdots p_s^{t_s}. \tag{9.1}$$

这个积称为*素数幂分解*(prime power decomposition)．例如，有

$$6=2^1 3^1,$$

⊖ 回想一下一个*素数* n 是一个大于 1 的且只有 1 和 n 两个正因子的正整数．参见 7.1.3 节．

$$12 = 3 \times 4 = 3^1 2^2,$$
$$80 = 16 \times 5 = 2^4 5^1,$$
$$60 = 4 \times 15 = 4 \times 3 \times 5 = 2^2 3^1 5^1.$$

定理 9.3 *假设 $n = p_1^{t_1} p_2^{t_2} \cdots p_s^{t_s}$ 是 n 的素数幂分解，$n>1$，且 r 是下面值中的最小数：*

$$(p_1^{t_1} - 1), (p_2^{t_2} - 1), \cdots, (p_s^{t_s} - 1).$$

那么存在 r 个秩为 n 的拉丁方的正交族.

我们在下面证明这一定理. 为了说明这一定理，注意，$12 = 2^2 3^1$. 于是有

$$2^2 - 1 = 3, \quad 3^1 - 1 = 2,$$

所以 $r=2$. 因此存在两个秩为 12 的正交拉丁方. 但这并没有说明不存在更多的 12×12 拉丁方的正交族. 注意，这里不能使用定理 9.2，因为 12 不是素数的幂.

让我们试尝着把定理 9.3 运用于 $n=6$. 我们有 $6 = 2^1 3^1$. 因为

$$2^1 - 1 = 1, \quad 3^1 - 1 = 2,$$

所以 $r=1$，且我们甚至无法从定理 9.3 得知是否存在一对正交的 6×6 拉丁方. 著名的数学家欧拉在 1782 年猜测不存在这样的对. 经历了 100 多年，这一猜测既没有得到证明也没有被推翻. 大约在 1900 年，Tarry 系统地考察了所有可能的 6×6 拉丁方对(一共有 812 851 200 对；但是通过使它们第一行变成 123456，只需要考虑 9408 对 6×6 拉丁方). 他成功地证明了欧拉的猜测是正确的. 因此，不存在秩为 6 的一对正交的拉丁方(参见 Tarry[1900, 1901]；对于现代的证明请参见 Stinson[1984]，这一证明没有考虑所有的情况).

下面是定理 9.3 的一个推论.

推论 9.3.1 *假设 $n>1$ 且或者 n 不能被 2 整除，或者 n 素数幂分解是*

$$n = 2^{t_1} p_2^{t_2} p_3^{t_3} \cdots,$$

其中 $t_1 > 1$. 那么存在一对秩为 n 的正交拉丁方.

证明 如果 $t_1 > 1$，那么

$$2^{t_1} - 1 \geqslant 3.$$

因为每一个其他的 p_i 都大于 2，所以

$$p_i^{t_i} - 1 \geqslant 2.$$

于是，对于 $r \geqslant 2$，结论成立. ■

推论 9.3.1 没有解决当秩为 $n=2k$，且 2 不能整除 k 时的正交拉丁方对的存在问题. 也是在 1782 年，欧拉猜测对于所有这样的秩 n 不存在一对秩为 n 的正交拉丁方. 对于 $n=2$，$n=6$ 来说他是正确的. 然而，与他通常的猜测相反，这次他错了. 直到 1960 年才证明了他是错误的.

定理 9.4(Bose, Shrikhande 和 Parker[1960]) *如果 $n>6$，$n=2k$ 且 2 不能整除 k，那么存在秩为 n 的一对正交拉丁方.*

现在我们可以把有关正交拉丁方对的存在问题加以概括.

定理 9.5 *对于所有 $n>1$，除 $n=2$ 和 $n=6$ 外，存在秩为 n 的一对正交拉丁方.*

因此，正交拉丁方对的存在问题已经完全解决. 对于更大的正交拉丁方族，情况并非如此. 对于 $n=2, 3, \cdots 9$，最大的 $n \times n$ 拉丁方正交族的大小是已知的. 因为通过定理 9.2 和定理 9.5，对于 $n=3, 4, 5, 7, 8, 9$，这个大小是 $n-1$，而对于 $n=2, 6$，这个大小为 1. 然而对于 $n=10$，我们甚至不知道是否存在 3 个两两正交的 $n \times n$ 拉丁方族.

【例 9.7 36 名官员问题】 欧拉遇到正交拉丁方的概念与实验设计无关，而与下面的问题有关. 有 36 名官员，在六个机构中的每一个有六个不同级别的 6 个官员. 寻找使用 6×6 方格来排列这 36 名官员的方法：这个 6×6 方格的每一行和每一列包含且只包含每个级别的一名官员，包含且只包含每个机构的一名官员，而且每个机构有且只有每个级别的一名官员. 能够做到这一点吗？必须这样排列这些官员，使得他们的级别形成一个拉丁方，并且还使得他们的机构形成一个拉丁方. 而且，级别和机构的序对出现一次且只能出现一次，所以这两个拉丁方必须是正交的. 现在，我们知道是做不到这一点的. ■

9.2.4 定理 9.3 的证明[⊖]

为了证明定理 9.3，我们首先证明下面的结果.

定理 9.6(MacNeish[1922]) *假设存在秩为 m 的 r 个拉丁方的正交族和秩为 n 的 r 个拉丁方的另一个正交族. 那么存在秩为 mn 的 r 个拉丁方的正交族.*

证明 设 $A^{(1)}$，$A^{(2)}$，…，$A^{(r)}$ 是秩为 m 的两两正交的拉丁方，而 $B^{(1)}$，$B^{(2)}$，…，$B^{(r)}$ 是秩为 n 的两两正交的拉丁方. 对于 $e=1, 2, \cdots, r$，设 $(a_{ij}^{(e)}, B^{(e)})$ 表示这样的 $n\times n$ 矩阵，其 u，v 项是序对 $(a_{ij}^{(e)}, b_{uv}^{(e)})$. 例如，假设 $A^{(1)}$，$A^{(2)}$，$B^{(1)}$，$B^{(2)}$ 如表 9.14 所示. 那么 $(a_{12}^{(1)}, B^{(1)})$ 和 $(a_{32}^{(2)}, B^{(2)})$ 就如此表中所示的那样. 设 $C^{(e)}$ 是如下所示的矩阵：

$$C^{(e)}=\begin{bmatrix}(a_{11}^{(e)},B^{(e)}) & (a_{12}^{(e)},B^{(e)}) & \cdots & (a_{1m}^{(e)},B^{(e)})\\ (a_{21}^{(e)},B^{(e)}) & (a_{22}^{(e)},B^{(e)}) & \cdots & (a_{2m}^{(e)},B^{(e)})\\ & & \cdots & \\ (a_{m1}^{(e)},B^{(e)}) & (a_{m2}^{(e)},B^{(e)}) & \cdots & (a_{mm}^{(e)},B^{(e)})\end{bmatrix}.$$

那么 $C^{(e)}$ 是一个 $mn\times mn$ 矩阵. 我们现在证明 $C^{(1)}$，$C^{(2)}$，…，$C^{(r)}$ 是秩为 mn 的拉丁方的正交族.

$$\langle(a_{ij}^{(e)},b_{uv}^{(e)}),(a_{ij}^{(f)},b_{uv}^{(f)})\rangle=\langle(a_{pq}^{(e)},b_{st}^{(e)}),(a_{pq}^{(f)},b_{st}^{(f)})\rangle.$$

表 9.14 正交拉丁方 $A^{(1)}$，$A^{(2)}$，$B^{(1)}$，$B^{(2)}$；矩阵 $(a_{12}^{(1)}, B^{(1)})(a_{32}^{(2)}, B^{(2)})$

$$A^{(1)}=\begin{array}{|ccc|}\hline 1&2&3\\2&3&1\\3&1&2\\\hline\end{array},\qquad A^{(2)}=\begin{array}{|ccc|}\hline 1&2&3\\3&1&2\\2&3&1\\\hline\end{array},$$

$$B^{(1)}=\begin{array}{|cccc|}\hline 4&3&2&1\\3&4&1&2\\2&1&4&3\\1&2&3&4\\\hline\end{array},\qquad B^{(2)}=\begin{array}{|cccc|}\hline 2&1&4&3\\4&3&2&1\\3&4&1&2\\1&2&3&4\\\hline\end{array},$$

$$(a_{12}^{(1)},B^{(1)})=\begin{bmatrix}(2,4)&(2,3)&(2,2)&(2,1)\\(2,3)&(2,4)&(2,1)&(2,2)\\(2,2)&(2,1)&(2,4)&(2,3)\\(2,1)&(2,2)&(2,3)&(2,4)\end{bmatrix},\quad (a_{32}^{(2)},B^{(2)})=\begin{bmatrix}(3,2)&(3,1)&(3,4)&(3,3)\\(3,4)&(3,3)&(3,2)&(3,1)\\(3,3)&(3,4)&(3,1)&(3,2)\\(3,1)&(3,2)&(3,3)&(3,4)\end{bmatrix}.$$

⊖ 本小节可以跳过.

为了证明 $C^{(e)}$ 是拉丁方，首先注意在给定的行中，不同列的两个项是由 $(a_{ij}^{(e)}, b_{uv}^{(e)})$ 和 $(a_{ik}^{(e)}, b_{uw}^{(e)})$ 给出的；所以它们不同，因为 $A^{(e)}$，$B^{(e)}$ 是拉丁方. 在给定的列中，不同行的两项是由 $(a_{ij}^{(e)}, b_{uv}^{(e)})$ 和 $(a_{kj}^{(e)}, b_{wv}^{(e)})$ 给出的；因为 $A^{(e)}$ 和 $B^{(e)}$ 是拉丁方，所以它们是不同的.

为了证明 $C^{(e)}$ 和 $C^{(f)}$ 是正交的，假设有

$$\langle(a_{ij}^{(e)},b_{uv}^{(e)}),(a_{ij}^{(f)},b_{uv}^{(f)})\rangle = \langle(a_{pq}^{(e)},b_{st}^{(e)}),(a_{pq}^{(f)},b_{st}^{(f)})\rangle.$$

于是有下面的等式：

$$(a_{ij}^{(e)},a_{ij}^{(f)}) = (a_{pq}^{(e)},a_{pq}^{(f)}),$$

所以，根据 $A^{(e)}$ 和 $A^{(f)}$ 的正交性，有 $i=p$ 且 $j=q$. 类似地，$B^{(e)}$ 和 $B^{(f)}$ 的正交性表明 $u=s$，$v=t$. ■

定理 9.4 的证明　根据定理 9.2，对于 $i=1, 2, \cdots, s$，有秩为 $p_i^{t_i}$ 的 $p_{i-1}^{t_i}$ 个拉丁方的正交族. 因此，对 $i=1, 2, \cdots, s$，存在秩为 $p_i^{t_i}$ 的 r 个拉丁方的正交族. 通过对 s 做数学归纳可以从定理 9.6 得到定理 9.3 的证明. ■

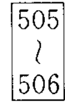
9.2.5　正交阵列及其在密码学中的应用[⊖]

假设 V 是一个 n 元素集合，且 $k\geqslant 2$ 是一个整数. 一个正交阵列(orthogonal array)OA(k, n)是一个 $n^2\times k$ 的矩阵 A，它的项是 V 的元素，并使得在 A 的任意两列内，满足 $a, b\in V$ 的每一个序对(a, b)，在 A 的某一行内正好出现一次. 表 9.15 给出一个 OA(4, 3)，其中 $V=\{1, 2, 3\}$，和一个OA(5, 4)，其中 $V=\{1, 2, 3, 4\}$.

表 9.15　一个 OA(4, 3)及一个 OA(5, 4)

$$\begin{bmatrix}2&2&2&2\\2&1&1&1\\2&3&3&3\\1&2&1&3\\1&1&3&2\\1&3&2&1\\3&2&3&1\\3&1&2&3\\3&3&1&2\end{bmatrix}\qquad\begin{bmatrix}1&4&4&4&4\\2&4&3&2&1\\2&3&4&1&2\\3&3&1&2&4\\4&4&1&3&2\\4&3&2&4&1\\1&1&1&1&1\\2&1&2&3&4\\1&3&3&3&3\\4&1&4&2&3\\3&4&2&1&3\\4&2&3&1&4\\1&2&2&2&2\\3&1&3&4&2\\2&2&1&4&3\\3&2&4&3&1\end{bmatrix}$$

正交阵列与拉丁方的正交族之间存在一个很简单的关系.



定理 9.7　*存在 OA(k, n)，当且仅当存在秩为 n 的 $k-2$ 个拉丁方的正交族.*

证明　不失一般性，假设正交阵列和拉丁方的项都来自于集合 $V=\{1, 2, \cdots, n\}$. 设 $A^{(1)}$，$A^{(2)}$，$\cdots$，$A^{(k-2)}$ 表示 $k-2$ 个秩为 n 的正交拉丁方. 构建矩阵 A 使其成为 n^2 个序列：

$$i,j,a_{ij}^{(1)},a_{ij}^{(2)},\cdots,a_{ij}^{(k-2)},$$

其中 $i, j\in\{1, 2, \cdots, n\}$，且 $a_{ij}^{(l)}$，或 $A^{(l)}(i, j)$ 是矩阵 $A^{(l)}$ 的 i，j 项. 为了看清 A 是一个 OA(k, n)，考虑 A 的列 u 和 v. 如果 $u=1$ 和 $v=2$，显然 $V\times V$ 的每一个序对(i, j)出现在这两列中及对应于(i, j)的行上. 如果 $u=1$ 或 2 且 $v\geqslant 3$，每一个序对(i, j)出现，因为 $A^{(v)}$ 的每列是$\{1, 2, \cdots, n\}$的一个排列. 最后，如果 u，$v\geqslant 3$，每一个序对(i, j)出现，因为 $A^{(u)}$ 和 $A^{(v)}$ 是正交的.

相反，假设 A 是 $V=\{1, 2, \cdots, n\}$ 上的 OA(k, n). 我们如下定义 $A^{(u)}$，$u\in\{1, 2, \cdots, k-2\}$. 给定 i 和 j，存在唯一使得 $i=A(r, 1)$，$j=A(r, 2)$的 r，于是设

$$a_{ij}^{(u)} = A(r,u+2).$$

⊖ 本小节大部分来自于 Stinson[2003].

不难证明如上定义的$A^{(1)}$，$A^{(2)}$，…，$A^{(k-2)}$是秩为n的正交拉丁方. 其证明留作练习(练习21). ■

【例9.8 密码学中的识别码】 在密码学中，涉及验证信息的真实性. 我们使用代码帮助我们加密这些信息，使它们很难被修改、偷窃或假冒. 关于代码的更多内容，参见第10章. 假设A向B发送一条信息. 在密码学理论中，我们通常把A称作Alice，而把B称为Bob. Alice通过电子邮件或传真或从她的移动电话把信息发出，这些都是不安全通道. Bob想要确认这一信息是否真是由Alice发送的，而且还要确认没有人改动了Alice发送的信息. 例如，这一信息可能是购买某样东西的一份订单，所以这个问题是电子商务的一个核心问题. 507

我们考虑一位名叫Oscar的外来者O干扰从Alice发送到Bob的信息的可能性. 假设Oscar能够简单地假冒Alice给Bob发送信息，或者Oscar能够更改由Alice发送的信息. Bob和Alice通过和信息一起发送一个识别码来防止Oscar的"攻击". 设M是可能信息的集合，C是识别符集合，而K是"钥匙"的集合. Alice和Bob在会面时或通过一个安全的通道，事先选定一把钥匙. 我们假设他们从K中随机选取钥匙. 与每一把钥匙$k\in K$相关联的是一个*识别规则*(authentication rule)r_k，它把一个识别符$r_k(m)\in C$指定给每一条信息$m\in M$. 如果Alice想要把信息m发送给Bob，那么她要发出信息(m, c)，其中$c=r_k(m)$. 当Bob接收到这条信息(m, c)时，事实上，他要检查c是否是$r_k(m)$. 如果不是，那么Bob有理由相信Oscar做了什么，而且他怀疑这条信息. 当然，也有这样的可能：Oscar正确地猜到了$r_k(m)$，因此这个过程不能察觉Oscar的所有进攻. 然而，如果发生这种情况的可能性很小而且与发送的实际信息无关，那么Alice和Bob将感到满意.

我们可以使用正交阵列构建识别规则. 假设$M=\{1, 2, \cdots, p\}$，$C=\{1, 2, \cdots, n\}$，$K=\{1, 2, \cdots, n^2\}$. 设矩阵A是一个OA(p, n)，且行由K的元素索引，而列由M的元素索引. 定义$r_k(m)=a_{km}$.

如果Oscar把一条信息(m, c)发送给Bob，那么$c=r_k(m)$的概率是多少呢？我们称这一概率是*假冒概率*(impersonation probability). 我们可以假设Oscar知道矩阵A，但是Oscar不知道正在使用哪一个$k\in K$. 给定m和c，那么A的n^2行中有n行i可能使得$a_{im}=c$. 因此，如果$c=r_k(m)$，那么Oscar选中某行i使得$r_i(m)$等于c的概率是$n/n^2=1/n$. 因此，这一代码使得Oscar每n次只有1次机会假冒Alice.

如果Oscar简单地用另一条信息(m', c')替换掉Alice发送的信息(m, c)，情况又如何呢？*欺骗概率*(deception probability)是使$c'=r_k(m')$的概率. 换句话说，欺骗概率是这样的概率，Bob将认为他接收到的信息是可信的，因此落入到Oscar的欺骗之中. Oscar看到了Alice发送的信息(m, c)，所以他知道$r_k(m)=c$，但是他不知道k的值. 他不得不希望$r_k(m')=c'$. 对于A的任意两列m和m'，序对(c, c')正好出现在A的某一行中的这两列上. 在A中存在n个行，这些行中c出现在列m上. 因此，如果Oscar随机从这些行中选出一行，那么在他选取的行中c'出现在列m上的几率是$1/n$. 因此，欺骗概率是$1/n$.

Alice和Bob所选择的识别码依赖于他们希望假冒概率和欺骗概率小到什么程度. 对于识别码的更多内容请参见Colbourn和Dinitz[1996]. ■ 508

【例9.9 门限方案和秘密分享】 存在这样的情形，某一项决策或行动非常敏感，需要一个小组里的若干成员的同意. 例如，发动核攻击的保密码，或银行里需要若干人共同打开的保险库等都属于这一情况.

假设 I 是 p 个人的集合，k 是启动某项行动(例如打开保险库或发动进攻)的密钥. $q\geqslant 2$ 是一个固定的整数，而且我们希望确认小组中的任意 q 个人可以一起决定 k，而小于 q 的任意子集都做不到这一点. 在较高的概率下实现这一功能的方法称为一个(q, p)*门限方案*((q, p)-threshold scheme). 固定一个钥匙集合 K 并确定一个不在 I 中的领导. 这位领导给每一个人关于 k 的部分信息，这些信息来自于部分信息的集合 P. 这位领导必须做到：任意 q 个人的部分信息足以计算出 k，而对于更小的子集的部分信息则不能做到这一点. 我们考虑 $q=2$ 的情况.

假设 $K=P=\{1, 2, \cdots, n\}$，且设 A 是一个正交阵列 $OA(p+1, n)$. 把 A 的前 p 列与参与者联系起来，而把最后一列与钥匙联系起来. 在这个小组里的所有人已知集合 A. 给定 $k\in K$，设 $R_k=\{i: a_{i,p+1}=k\}$是由最后一项为 k 的那些行组成的集合. 这位领导者随机选出某行 $i\in R_k$，并把部分信息 a_{iu} 给第 u 个人.

人 u 和人 v 能决定钥匙 k 吗？假设 u 得到部分信息 p_u 而 v 得到部分信息 p_v. 因为存在唯一一行 i，使得 $a_{ru}=p_u$，$a_{rv}=p_v$，所以 u 和 v 能决定 r，因此能够寻找到 $a_{r,p+1}$，这就是他们所需要的钥匙 k.

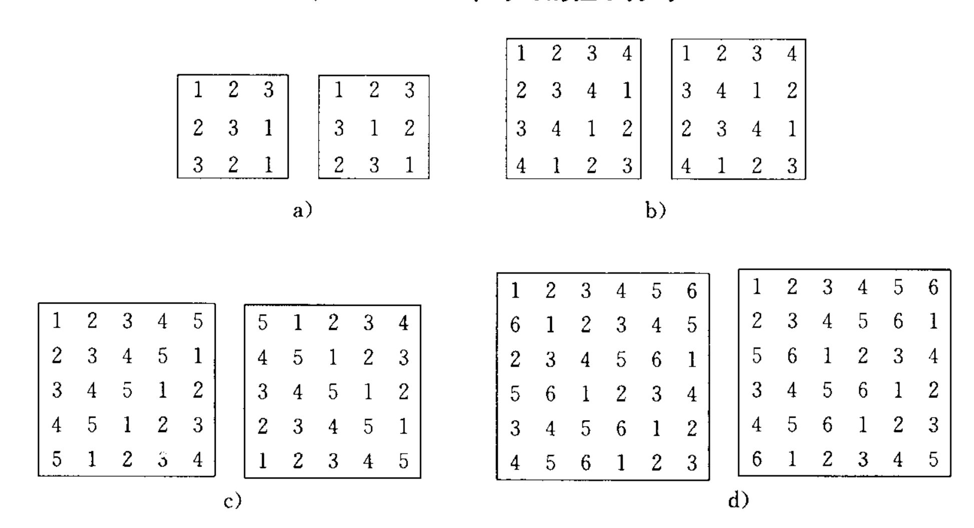
任何一个人 u 只基于他的部分信息 p_u 可以决定 k 吗？对于钥匙的任意可能值 k'，存在唯一的行 i，对于这一行有 $a_{iu}=p_u$，$a_{i,p+1}=k'$. 人 u(如果不与其他人分享信息)没有办法知道 n 个可能行 i 中的哪行是正确的(即被领导者选中的那行). 因此，只基于自己的部分信息，u 正确地猜测出钥匙的概率是 $1/n$. 因此，我们找到了能以较高的概率完成工作的$(2, p)$门限方案.

关于门限方案和秘密分享的更多内容可以参见 Colbourn 和 Dinitz[1996]. ■

本节练习

1. 对于表 9.16 中的每一对拉丁方，确定它们是否是正交的.

表 9.16　9.2 节练习的拉丁方对



a)

1	2	3
2	3	1
3	2	1

1	2	3
3	1	2
2	3	1

b)

1	2	3	4
2	3	4	1
3	4	1	2
4	1	2	3

1	2	3	4
3	4	1	2
2	3	4	1
4	1	2	3

c)

1	2	3	4	5
2	3	4	5	1
3	4	5	1	2
4	5	1	2	3
5	1	2	3	4

5	1	2	3	4
4	5	1	2	3
3	4	5	1	2
2	3	4	5	1
1	2	3	4	5

d)

1	2	3	4	5	6
6	1	2	3	4	5
2	3	4	5	6	1
5	6	1	2	3	4
3	4	5	6	1	2
4	5	6	1	2	3

1	2	3	4	5	6
2	3	4	5	6	1
5	6	1	2	3	4
3	4	5	6	1	2
4	5	6	1	2	3
6	1	2	3	4	5

2. 验证表 9.11 的三个拉丁方形成一个正交族.
3. 对于表 9.17 中的每一个拉丁方族，确定它是否是正交的.
4. 假设 A 是一个 $n\times n$ 拉丁方. 对于下面的各操作，确定其结果是否产生一个新的拉丁方.
   (a)在 2 和 4 出现的任何地方互换 2 和 4.
   (b)用每一行的逆置替换该行.
   (c)用 A 的转置替换 A.

509

表 9.17 9.2 节练习的拉丁方族

1	3	2	4
3	1	4	2
2	4	1	3
4	2	3	1

3	1	4	2
2	4	1	3
1	3	2	4
4	2	3	1

2	4	1	3
1	3	4	2
3	1	4	2
4	2	3	1

a)

1	2	3	4	5
2	3	4	5	2
3	4	5	1	2
4	5	1	2	3
5	1	2	3	4

1	2	3	4	5
3	4	5	1	2
5	1	2	3	4
2	3	4	5	1
4	5	1	2	3

1	2	3	4	5
5	1	2	3	4
4	5	1	2	3
3	4	5	1	2
2	3	4	5	1

b)

5. 对于下列各数 n，你能够确定存在秩为 n 的 3 个拉丁方的正交族吗？为什么？
   (a) $n=12$　(b) $n=17$　(c) $n=21$　(d) $n=24$
   (e) $n=33$　(f) $n=36$　(g) $n=55$　(h) $n=75$
   (i) $n=161$　(j) $n=220$　(k) $n=369$　(l) $n=539$
6. 如果 $n=539$，证明存在秩为 n 的 10 个两两正交的拉丁方集合.
7. 如果 $n=130$，存在秩为 n 的正交拉丁方对吗？为什么？
8. 是否存在秩为 12 的正交拉丁方对？为什么？
9. 如果 n 能被 4 整除，你能否(利用我们给出的定理)肯定是否存在三个秩为 n 的两两正交的拉丁方集合？给出你的理由.
10. 对于表 9.18 所示的秩为 5 的拉丁方正交族，利用定理 9.1 的证明过程重新排列元素使得每一个拉丁方的第一行是 12345.

表 9.18 拉丁方的正交族

5	1	2	3	4
4	5	1	2	3
3	4	5	1	2
2	3	4	5	1
1	2	3	4	5

4	5	1	2	3
2	3	4	5	1
5	1	2	3	4
3	4	5	1	2
1	2	3	4	5

3	4	5	1	2
5	1	2	3	4
2	3	4	5	1
4	5	1	2	3
1	2	3	4	5

2	3	4	5	1
3	4	5	1	2
4	5	1	2	3
5	1	2	3	4
1	2	3	4	5

11. 假设两个正交的 8×8 拉丁方都以 87654321 作为第一行.
    (a)它们是否有可能有相同的 2，4 项？
    (b)关于每一个拉丁方都以 87654321 作为第一行的 8×8 两两正交拉丁方的可能数量，你的答案告诉你什么信息？
12. 假设两个正交的 7×7 拉丁方都以 1234567 作为最后一行.
    (a)这两个拉丁方能有相同的 1，3 项吗？
    (b)关于每一个拉丁方都以 1234567 作为最后一行的 7×7 两两正交拉丁方的可能数量，你的答案告诉你什么信息？
13. 假设两个正交的 4×4 拉丁方都以 1234 作为它们的主对角线.
    (a)它们能有相同的 2，3 项吗？
    (b)关于每一个拉丁方都以 1234 作为主对角线的 4×4 两两正交拉丁方的可能数量，你的答案告诉你什么信息？

510 ~ 511

14. 利用表 9.14 的拉丁方寻找秩为 12 的正交拉丁方对.
15. 寻找秩为 9 的正交拉丁方对.
16. 如果存在秩为 n 的正交拉丁方对，且 A 是秩为 n 的拉丁方，A 不一定是某个正交拉丁方对中的成员. 给出一个对此加以解释的例子.

17. 给定表 9.15 的 OA(4, 3)，根据该数组寻找两个正交拉丁方.
18. 给定表格 9.15 的 OA(5, 4)，根据这个阵列寻找三个两两正交的拉丁方.
19. 给出表 9.10 的两个正交的拉丁方，寻找对应的正交阵列 OA(2, 4).
20. 给定表 9.13 的三个拉丁方的正交族，寻找对应的正交阵列 OA(5, 4).
21. 完成定理 9.7 的证明.
22. 对于下列的哪一组 p 和 n 的值，存在 OA(p, n)?
    (a) $p=3$, $n=81$　　(b) $p=4$, $n=6$　　(c) $p=4$, $n=63$
23. (Stinson[2003])
    (a) 如果 Alice 与 Bob 传送 200 条可能的信息，而且要确保把 Oscar 所做的假冒概率限制到小于 1/1000，解释一个正交阵列 OA(200, 1009)是如何帮助他们的.
    (b) 证明存在这样的正交阵列.
    (c) 解释为什么在 OA(200, 1009)中不使用小于 1009 的数?
24. 在例 9.8 中，假设 Oscar 有把可能的钥匙限制到一个大小为 s 的固定子集上的信息. 这一信息是如何改变假冒概率的?
25. (Stinson[2003])
    (a) 如果一个 10 人小组想要建立一个(2, 10)门限方案，并保证任何人能够猜测到这个密钥的概率小于 1/100，解释一个正交阵列 OA(11, 100)是如何实现这一点的.
    (b) 证明存在这样的正交阵列.
26. (Stinson[1990, 2003])在例 9.8 中，假设使用识别规则的方法，但是不假设这一规则是由正交阵列决定的. 证明假冒概率至少总是 $1/|C|$，而且这个概率等于 $1/|C|$ 当且仅当对于所有 $m\in M$, $c\in C$ 有 $|\{k: r_k(m)=c\}|=|K|/|C|$.
27. (Stinson[1990, 2003])对于练习 26 的情况，假设假冒概率是 $1/|C|$. 证明欺骗概率至少是 $1/|C|$，而且它等于 $1/|C|$ 当且仅当对于所有 $m, m'\in M$, C, $c'\in C$ 有 $|\{k: r_k(m)=c\}\cap\{k: r_k(m')=c'\}|=|K|/|C|$.
28. (Stinson[1990, 2003])对于练习 26 的情况，证明如果 $|M|=p$, $|C|=n$，且如果假冒概率和欺骗概率都等于 $1/n$，那么 $|K|\geqslant n^2$，且 $|K|=n^2$ 当且仅当识别规则定义一个正交阵列 OA(p, n)的行(这就是证明正交阵列给出最小化所需钥匙数量的识别规则).

512

9.3 有限域与拉丁方族[⊖]

本节的目标是给出定理 9.2 的一个构造性证明，也就是说，如果 $n>1$，且对于某个素数 p, $n=p^k$，那么存在秩为 n 的完全正交拉丁方族. 我们先从数学预备知识开始.

9.3.1 模算术

只有有限多个数的算术是组合设计构造的基础. 它们在计算中也很重要，其中，所考虑的整数集合的大小存在实际界限. 本小节介绍只有有限多个元素的算术的简单例子，即模算术. 在 9.3.3 节中，我们介绍只有有限多个元素的算术的一般概念，即有限域的概念. 然后我们使用有限域构造完全正交拉丁方族. 模算术和有限域构成计算机上取一个位串且生成另一个位串的位移寄存器操作的基础. 关于此应用的讨论，可以参见 Fisher[1977].

我们考虑一个整数除以 3 时的余数. 我们发现

$$
\begin{aligned}
0 &= 0\times 3+0, & 1 &= 0\times 3+1, & 2 &= 0\times 3+2,\\
3 &= 1\times 3+0, & 4 &= 1\times 3+1, & 5 &= 1\times 3+2,\\
6 &= 2\times 3+0, & 7 &= 2\times 3+1, & 8 &= 2\times 3+2,\\
9 &= 3\times 3+0, & 10 &= 3\times 3+1, & 11 &= 3\times 3+2.
\end{aligned}
$$

⊖ 跳过本节不会失去连续性. 作为一种选择，读者可以阅读 9.3.5 节之外的所有内容.

余数总是0，1，2中的一个．我们说两个整数 a 和 b 模3同余(congruent modulo 3)，写作 $a\equiv b(\text{mod } 3)$，如果它们整除3时余数相同．例如，$2\equiv5(\text{mod } 3)$，$1\equiv7(\text{mod } 3)$．一般地，如果 a，b 和 n 都是整数，我们说 a 与 b 模 n 同余，写作 $a\equiv b(\text{mod } n)$，如果 a 和 b 整除 n 时余数相同．例如，$29\equiv17(\text{mod } 4)$，因为 $29=7\cdot4+1$ 而 $17=4\cdot4+1$．我们每天看钟时使用的就是模12同余．钟的指针表示时间模12．类似地，汽车的里程计给出汽车跑过的里程数模1 000 000(这个数的选取依赖于汽车的品牌和型号)．

尽管模算术很完美且很有效，但是软件系统的一个主要的麻烦就是由它所引起的．千年虫问题(Y2K)的发生就归因于计算机软件的内在工作方式．很多软件开发者在记录年代时使用两位数字指示器；前提假设是20世纪．例如，使用2/17/59中的59表示1959年．因此，这两个数字是模100．在2000年当Y2K问题被认清时，它才为众人所知，当使用00时，软件无法区分它表示的是1900年还是2000年．

513

现在我们固定一个数 n，并考虑整数的集合 $Z_n=\{0, 1, 2\cdots, n-1\}$．每一个整数都与集合 Z_n 中的一个整数模 n 同余．如果我们把集合 Z_n 中的两个整数相加，那么它们的和不一定还在 Z_n 中．然而它们的和与 Z_n 中的一个元素同余．在进行*模加法*(modular addition)时，如果 a 和 b 在 Z_n 中，我们定义 $a+b$ 是 Z_n 中的一个数，它与 a 和 b 的一般和模 n 同余．例如，假设 $n=3$，那么 $2+2=4$ 是一般算术的和，而这个和与1同余模3．因此在模3加法中，$2+2$ 是1．类似地，$1+2$ 是0，$1+1$ 是2．在模4加法中，$3+3$ 是2，$2+2$ 是0．模乘法的运算方式类似．如果 a，b 在 Z_n 中，我们定义 $a\times b$ 是 Z_n 中的一个整数，这个整数与 a 和 b 的一般积模 n 同余．例如，在模3乘法中，2×2 等于1，因为2与2的一般积是4，而4与1模3同余．类似地，在模4乘法中，3×3 等于1，2×2 等于0．

我们可以通过给出加法表和乘法表概括 Z_n 上的*运算*(operation)$+$和$\times$．表9.19和表9.20分别给出了 $n=2$ 和 $n=3$ 时的加法表和乘法表．在这些表中，Z_n 的元素以相同的顺序列在行和列上，而且对应于 a 的行、b 列处的项是 $a+b$ 或 $a\times b$，取决于表．

表9.19 Z_2 上模2的加法和乘法运算

+	0	1
0	0	1
1	1	0

×	0	1
0	0	0
1	0	1

表9.20 Z_3 上模3的加法和乘法运算

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

×	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

9.3.2 模算术和RSA密码系统[⊖]

在因特网上发送敏感信息(例如，信用卡号码)使密码受到了很大的关注，所谓的密码是以隐藏信息为目的的代码(一旦信息被隐藏，它就可以被编码以保证正确传输并解码．编码理论，特别是错误校正和错误检测代码是第10章的课题)．我们将描述"公钥密码系统"，这一系统是基于Diffie和Hellman[1976]的工作，由Rivest，Shamir和Adleman[1978]所创建的，现在被称为RSA．RSA已在7.1.3节中讨论过．

514

假设你需要把你的信用卡号码通过因特网传送给一家零售店．那么你将如何改变这一号码(加密)，使得当这一传输被其他人读取时，无法获取这一号码呢？这一问题的答案要以素数和模算术为基础．

首先，我们需要一个关于素数的定理．

定理9.8(费马小定理) *如果 p 是一个素数，x 是一个正整数，而且 $x\not\equiv0$(模 p)，那么*

⊖ 本小节取自Hill[1991]，可以跳过．

$$x^{p-1} \equiv 1(\bmod p).$$

证明⊖ 考虑集合 $A=\{1x, 2x, 3x, \cdots, (p-1)x\}$. A 的元素模 p 互不相同(为了弄清这一点，假设下面的内容不成立，即对于 $1\leqslant i\neq j\leqslant p-1$，有 $ix\equiv jx(\bmod p)$，因为 $x\not\equiv 0(\bmod p)$，所以 x^{-1}存在，于是用 x^{-1}乘以等式的两边得到 $i\equiv j(\bmod p)$，这意味着 $i=j$，因为 $1\leqslant i, j<p$；这是矛盾的). 同样，因为 $x\not\equiv 0(\bmod p)$，所以 A 的每一个元素 $ix\not\equiv 0(\bmod p)$，因为 $1\leqslant i\leqslant p-1$. 因此，$\{1x(\bmod p), 2x(\bmod p), \cdots, (p-1)x(\bmod p)\}=\{1, 2, \cdots, p-1\}$. 因此有下式：

$$(1x)(2x)\cdots((p-1)x) \equiv (1)(2)\cdots(p-1)(\bmod p),$$

或者

$$(p-1)!x^{p-1} \equiv (p-1)!(\bmod p).$$

显然，$(p-1)! \not\equiv (\bmod p)$，所以$[(p-1)!]^{-1}$存在. 在上式两边同时乘以$[(p-1)!]^{-1}$完成证明. ■

为了使用 RSA，商店选择两个不同且非常大的素数 p 和 q. 然后计算 $r=pq$，并确定两个正整数 s 和 t 使得有

$$st \equiv 1(\bmod (p-1)(q-1)),$$

即使得对于某个整数 j 有 $st=j(p-1)(q-1)+1$. 于是商店公开 r 和 s(因此就有了术语公钥(public-key)). 作为一个例子，假设商店选择了素数 $p=37$ 和 $q=23$. 这时，$r=37 \cdot 23=851$，而这家商店可以选择 $s=5$ 和 $t=317$，因为有：

$$\begin{aligned} 5 \cdot 317 = 1585 &= 2(36)(22)+1 = 2(37-1)(23-1)+1 \\ &\equiv 1[\bmod (37-1)(23-1)]. \end{aligned}$$

(选择其他 s 和 t 也是可以的. 然而，应该避免选择 1.)重要的是要注意到商店不公开 p，q 或 t，而只公开 r 和 s. 515

在因特网上传输你的信用卡号码之前，为了加密你的信用卡号码 n，你要计算并传输：

$$m \equiv n^s(\bmod r). \tag{9.2}$$

此时出现两个问题：(1)如果某人偷窃了这个数 m，那么他们如何确定你的信用卡号码 n？(2)当商店接受到这个数 m，那么他们如果何计算(解密)你的信用卡号码 n？这两个问题都很重要.

商店计算 $m^t(\bmod r)$并利用费马小定理(定理 9.8)确定你的信用卡号码 n. 注意，为了解密，实质上，商店只需要 t! 因为根据等式(9.2)有 $m^t\equiv n^{st}(\bmod r)$和 $n^{st}\equiv n^{j(p-1)(q-1)+1}$. 接下来，如果 $n\not\equiv 0(\bmod p)$，那么根据费马小定理(定理 9.8)，有 $n^{j(p-1)(q-1)+1}=(n^{p-1})^{j(q-1)}n\equiv n(\bmod p)$. 如果 $n\equiv 0(\bmod p)$，那么一定有 $n^{j(p-1)(q-1)+1}\equiv n(\bmod p)$. 无论哪种情况，$p$ 一定是 $n^{j(p-1)(q-1)+1}-n$ 的因子. 类似地，q 也一定是 $n^{j(p-1)(q-1)+1}-n$ 的因子. 因为 p 和 q 是不同的素数，且有 $r=pq$，所以

$$n^{j(p-1)(q-1)+1} \equiv n(\bmod r) = n,$$

这就是你的信用卡号码，只要你的信用卡号码不大于 r. 因为这个系统的安全是基于非常大的 r 的使用的，所以 $n>r$ 的情况不会出现.

事实上，整个系统的安全性依赖于 t 的无效性. 因为 r 和 s 是公开的，为什么想偷你信用卡号码的人不能从它们推断出 t 呢？答案在于事实上这需要原始素数 p 和 q 的信息，即分解成 $r=pq$ 的能力. 然而，素数因子分解是一个难题. 事实上，根据 2.18 节中的表记法，它是典型的 NP 完全问题. 对于我们的例子，$r=851$ 的因子分解不需要太长的时间，但是如果素数 p 和 q 的每一个都是 100 位数，那么即便是使用当今(2003 年)最好的算法和最快的计算机分解大约有 200 位数的 r 也是不可行的(Stinson[2003]). 所以，只要没有发现复杂度是多项式的素数因子分

⊖ 这一证明可以跳过. 这一证明利用了在此没有详细阐述的著名的模算术的性质. 另一种证明见练习 9.

解的确定性算法(且信用卡号码不得超过200位)，那么RSA密码系统就是安全的[1]. 更多关于密码和RSA的内容，可以参见Garrett[2001]、Joye和Quisquater[1998]、Kaliski[1997]、Koblitz[1994]、Menezes，van Oorschot和Vanstone[1997]、Salomaa[1996]和Sloane[1981].

9.3.3 有限域 GF(p^k)

接下来，我们考虑9.3.1节所介绍的模算术的扩展. 假设X是一个集合. X上的一个二元运算(binary operation)o是一个函数，它给X上的每一个元素序对指定X的另一个元素，通常记为aob. 例如，如果X是整数集合，那么$+$和$\times$定义X上的二元运算. 如果X是一个有限集合，我们能够通过给定一个诸如表9.19和表9.20那样的表定义X上的二元运算o. 516

域(field)$\mathcal{F}$是一个三元组(F，$+$，$\times$)，其中F是一个集合，而$+$和$\times$是F上的二元运算(不一定是通常的$+$和$\times$运算)，且有某些特定的条件成立. 这些条件如下[2]：

条件 **F1**(封闭性)[3]. 对于F中的所有a，b，$a+b$和$a\times b$都在F中.

条件 **F2**(结合性). 对于F中的所有a，b，c，有

$$a+(b+c)=(a+b)+c,$$
$$a\times(b\times c)=(a\times b)\times c.$$

条件 **F3**(交换性). 对于F中的所有a，b，有

$$a+b=b+a,$$
$$a\times b=b\times a.$$

条件 **F4**(单位元).

(a)F中存在一个元素，记为0并称为加法单位元，使得对于F中所有a，有

$$a+0=a.$$

(b)F中存在一个不同于0的元素，记为1并称为乘法单位元，使得对于F中的所有a，有

$$a\times 1=a.$$

条件 **F5**(逆).

(a)对于F中的所有a，F中存在一个元素b，称为a的加法逆，使得有

$$a+b=0.$$

(b)对于F中的所有a，且$a\neq 0$，F中存在一个元素b，称为a的乘法逆，使得有

$$a\times b=1.$$ 517

条件 **F6**(分配性). 对于F中的所有a，b，c，有

$$a\times(b+c)=(a\times b)+(a\times c).$$

[条件**F1**、**F2**、**F4**和**F5**说明(F，$+$)在第8章的意义下是一个群. 同时，如果$F'=F-\{0\}$，那么上面这些条件也说明(F'，$\times$)是一个群，因为我们能够证明如果$a\neq 0$或$b\neq 0$，那么$a\times b$不等于0，而且我们还能证明a的乘法逆不等于0]. 注意，从条件**F1**～**F6**可以证明一个元素a的加法逆和乘法逆分别是唯一的；它们分别记作$-a$和a^{-1}.

下面是一些域的例子.

---

[1] 在7.1.3节，我们注意到近来的工作已经证明验证一个整数是否是素数的问题可以高效地完成(即在多项式时间内完成). 尽管这一结果不能说明把一个数有效地因子分解成素数的可能，但是它的确引起关于这样的因子分解是否可能的争论. 这是一个关键问题，因为因子分解的困难程度是密码学的核心问题.

[2] 我们尽量简单地讨论域. 没有这方面知识背景的读者可以参见Dornhoff和Hohn[1978]、Durbin[1999]、Fisher[1977]、Gilbert和Gilbert[1999]上的初步讨论.

[3] 实际上，条件**F1**在运算的定义下是显然的.

1.(Re, +, ×),其中 Re 是实数集合,而+和×就是通常的加法和乘法运算.

2.(Q, +, ×),其中 Q 是有理数的集合,而+和×就是通常的加法和乘法运算.

3.(C, +, ×),其中 C 是复数的集合,而+和×是通常的加法和乘法运算.

然而,(Z, +, ×),其中 Z 是整数的集合,而+和×是通常的加法和乘法运算,不是一个域.条件 **F1**~**F4** 和条件 **F6** 成立,但是条件 **F5** 的(b)不成立. Z 中不存在 b 使得 $2\times b=1$.

现在,我们考虑一些有限域(finite field)的例子,即 F 是一个有限集合的域.考虑(Z_2, +, ×),其中+和×是模 2 的运算.那么容易验证,它是一个域.注意,0 和 1 分别是加法和乘法的单位元. 1 的加法逆是 1,因为 $1+1=0$,1 的乘法逆是 1,因为 $1\times1=1$.

类似地,(Z_3, +, ×)是域,其+和×是模 3 的.加法和乘法的单位元分别还是 0 和 1.注意 2 的加法逆是 1,因为 $2+1=0$. 2 的乘法逆是 2,因为 $2\times2=1$.

在模加法和模乘法下,Z_n 总是域吗?这一问题的答案是否定的.考虑 Z_6.我们有 $3\times2=0$.如果 Z_6 在模 6 的加法和乘法下是域,设 2^{-1} 表示 2 的乘法逆.我们有

$$(3\times2)\times2^{-1}=0\times2^{-1}=0.$$

然而,

$$(3\times2)\times2^{-1}=3\times(2\times2^{-1})=3\times1=3.$$

我们得出 $0=3$ 的结论,这是矛盾的.

定理 9.9 *对于 $n\geq2$,当且仅当 n 是素数时 Z_n 在模 n 的加法和乘法运算下是域.*

定理 9.9 的证明留作练习(练习 16).

我们通过下面的问题来结束本小节:对于什么样的 n,存在 n 个元素的有限域?我们将给出一个明确的回答.注意,根据定理 9.9,Z_4 在模加法和乘法下不定义一个域.然而,可以在{0, 1, 2, 3}上定义加法和乘法,使其形成一个域.例如,表 9.21 给出了这样的运算.这些运算定义域的验证留作练习(练习 17)(熟悉代数的读者可以通过设 $2=w$ 和 $3=1+w$,并在下面定义的 2 元素有限域 $GF[2]$ 上做模不可约多项式 $1+w+w^2$ 的加法和乘法,就可以推导出这些表).实际上用于很多大型计算机上的二进制数的算术,都是基于某个 n 的 Z_{2^n} 的(这里,$n=2$).关于这一特殊算术的讨论可以参见 Dornhoff 和 Hohn[1978]、Hennessy 和 Patterson[1998, Ch. 4]、Patterson 和 Hennessy[1998, Appendix A].

518

表 9.21 四元素的域 $GF(2^2)$ 的加法和乘法表

+	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

×	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

如果 $n=6$,不可能在 n 个元素的集合如{0, 1, 2, 3, 4, 5}上定义加法和乘法,使其成为一个有限域.这一情况概括为定理 9.10.

定理 9.10 *如果(F, +, ×)是一个有限域,那么存在素数 p 和正整数 k 使得 F 有 p^k 个元素.反之,对于所有素数 p 和正整数 k,存在有 p^k 个元素的有限域.*

我们可以在大多数现代代数学的书中找到这一定理证明:例如,上一页第二个脚注中的任意参考资料中都有此定理的证明.可以证明对于素数 p 和正整数 k,在同构的意义下[⊖],存在唯一一个有 p^k 个元素的域,这唯一的 p^k 个元素的域记为 GF(p^k)(字母 GF 表示伽罗瓦域,是以著名数学家 Evariste Galois 的名字命名的,他为现代代数学的发展奠定了基础).

⊖ 两个域$\mathcal{F}$和$\mathcal{G}$同构,如果存在从$\mathcal{F}$到$\mathcal{G}$上保持加法和乘法的一一映射.

9.3.4 当 n 是素数幂时，$n\times n$ 拉丁方的完全正交族的构造

现在，我们给出 $n\times n$ 拉丁方的完全正交族的构造方法，对于素数 p 和正整数 k，及 $n>1$，这一方法适用于 $n=p^k$ 时的情况. 这将证明定理9.2. 设 b_1，b_2，…，b_n 是 $n=p^k$ 个元素的有限域 GF(n)的元素. 设 b_1 是这个域的乘法单位元，而 b_n 是加法单位元. 对于 $e=1$，2，…，$n-1$，通过下式定义 $n\times n$ 数组 $A^{(e)}=(a_{ij}^{(e)})$： 519

$$a_{ij}^{(e)}=(b_e\times b_i)+b_j, \tag{9.3}$$

其中＋和×是域 GF(n)的运算. 在9.3.5节，我们将证明 $A^{(e)}$ 是一个拉丁方，且 $A^{(1)}$，$A^{(2)}$，…，$A^{(n-1)}$ 是正交族. 因此，如果 $n>1$，我们就得到 $n\times n$ 拉丁方的一个完全正交族. 例如，如果 $n=3$，我们利用 GF(3)，其加法和乘法运算是如表9.20定义的模3加法和乘法. 我们设 $b_1=1$，$b_2=2$，$b_3=0$（记住，b_1 被选作乘法单位元，b_n 被选作加法单位元）. 于是，我们发现 $A^{(1)}$ 和 $A^{(2)}$ 由表9.22给出. 例如，为了弄明白 $A^{(2)}$ 的项1，2是如何计算的，注意

表9.22 根据(9.3)式由有限域 GF(3)定义的正交拉丁方 $A^{(1)}$ 和 $A^{(2)}$

$$A^{(1)}=\begin{bmatrix}2&0&1\\0&1&2\\1&2&0\end{bmatrix}\qquad A^{(2)}=\begin{bmatrix}0&1&2\\2&0&1\\1&2&0\end{bmatrix}$$

$$(b_2\times b_1)+b_2=(2\times 1)+2=2+2=1.$$

很容易直接验证表9.22的 $A^{(1)}$ 和 $A^{(2)}$ 是拉丁方，且它们是正交的.

作为另一个例子，假设 $n=4$. 于是，我们利用 GF(4)=GF(2^2)，其运算是表9.21给出的加法和乘法. 取 $b_1=1$，$b_2=2$，$b_3=3$，$b_4=0$，并利用(9.3)式我们得到表9.23所示的三个两两正交的拉丁方. 为了看清这些项是如何得到的，注意

$$a_{23}^{(3)}=(b_3\times b_2)+b_3=(3\times 2)+3=1+3=2,$$

在上面的计算中，我们使用了表9.21的加法和乘法规则. 520

表9.23 由表9.21的有限域 GF(2^2)得到的4×4拉丁方正交族

$$A^{(1)}=\begin{bmatrix}0&3&2&1\\3&0&1&2\\2&1&0&3\\1&2&3&0\end{bmatrix}\quad A^{(2)}=\begin{bmatrix}3&0&1&2\\2&1&0&3\\0&3&2&1\\1&2&3&0\end{bmatrix}\quad A^{(3)}-\begin{bmatrix}2&1&0&3\\0&3&2&1\\3&0&1&2\\1&2&3&0\end{bmatrix}$$

9.3.5 当 $n=p^k$ 时，完全正交族的构造的证明㊀

为了证明9.3.4节的构造方法，我们首先一般地证明如果 $A^{(e)}$ 是由(9.3)式定义的，那么它是拉丁方. 假设 $a_{ij}^{(e)}=a_{ik}^{(e)}$. 于是

$$(b_e\times b_i)+b_j=(b_e\times b_i)+b_k. \tag{9.4}$$

在(9.4)的等式两边加上($b_e\times b_i$)的加法逆 c，并使用加法的结合性和交换性，我们得到

$$\begin{gathered}(c+[(b_e\times b_i)+b_j])=(c+[(b_e\times b_i)+b_k]),\\ ([c+(b_e\times b_i)]+b_j)=([c+(b_e\times b_i)]+b_k),\\ 0+b_j=0+b_k,\\ b_j=b_k,\\ j=k.\end{gathered}$$

因此，同一行的所有元素都是不相同的.

接下来，假设 $a_{ji}^{(e)}=a_{ki}^{(e)}$. 于是，

㊀ 本小节可以跳过.

$$(b_e \times b_j) + b_i = (b_e \times b_k) + b_i. \tag{9.5}$$

在等式(9.5)的两边加上 b_i 的加法逆，并利用加法的结合性，我们得到

$$b_e \times b_j = b_e \times b_k. \tag{9.6}$$

现在，我们用 b_e 的乘法逆 a 乘以等式(9.6)的两边，因为 $b_e \neq 0$，所以它的乘法逆存在，并利用×运算的交换性和结合性，我们得到

$$a \times (b_e \times b_j) = a \times (b_e \times b_k),$$

$$(a \times b_e) \times b_j = (a \times b_e) \times b_k,$$

$$1 \times b_j = 1 \times b_k,$$

$$b_j = b_k,$$

521

$$j = k.$$

因此，同一列的所有元素都是不相同的，于是我们得出 $A^{(e)}$ 是拉丁方的结论.

最后，我们证明对于 $e \neq f$，$A^{(e)}$ 和 $A^{(f)}$ 是正交的. 假设

$$(a_{ij}^{(e)}, a_{ij}^{(f)}) = (a_{kl}^{(e)}, a_{kl}^{(f)}).$$

那么

$$a_{ij}^{(e)} = a_{kl}^{(e)} \quad 和 \quad a_{ij}^{(f)} = a_{kl}^{(f)},$$

所以

$$(b_e \times b_i) + b_j = (b_e \times b_k) + b_l, \tag{9.7}$$

且

$$(b_f \times b_i) + b_j = (b_f \times b_k) + b_l. \tag{9.8}$$

使用域的性质，我们可以从(9.7)式减去(9.8)式；即我们把(9.8)式两边的加法逆分别加到(9.7)式的两边. 于是得到

$$(b_e \times b_i) - (b_f \times b_i) = (b_e \times b_k) - (b_f \times b_k),$$

上式的一的意思是加上加法逆. 因此，再一次利用域的性质，我们得到

$$(b_e - b_f) \times b_i = (b_e - b_f) \times b_k. \tag{9.9}$$

最后，因为 $e \neq f$，因此有 $(b_e - b_f) \neq 0$，所以 $(b_e - b_f)$ 有乘法逆. 用这个乘法逆乘以等式(9.9)的两边，我们得到等式

$$b_i = b_k,$$

因此

$$i = k.$$

此时，(9.7)式给出下面的等式：

$$(b_e \times b_i) + b_j = (b_e \times b_i) + b_l,$$

从这一等式，我们得到

$$b_j = b_l,$$

$$j = l.$$

因此有 $i=k$ 和 $j=l$，所以我们得出结论 $A^{(e)}$ 和 $A^{(f)}$ 是正交的. 这就完成了 $A^{(1)}$，$A^{(2)}$，…，$A^{(n-1)}$ 是 $n \times n$ 拉丁方的正交族的证明.

本节练习

1. 对于下列每一组 a 和 n 的值，确定{0，1，2，…，$n-1$}中的一个数 b，使得 $a \equiv b \pmod n$.

   (a)$a=37$，$n=5$ (b)$a=42$，$n=3$ (c)$a=8$，$n=10$

(d)$a=11$，$n=9$　　(e)$a=625$，$n=71$　　(f)$a=1652$，$n=7$

522

2. 对于下列每一组 a，b 和 n 的值，利用模 n 的加法和乘法计算 $a+b$ 和 $a\times b$.

   (a)$a=2$，$b=4$，$n=4$　　(b)$a=4$，$b=5$，$n=12$　　(c)$a=5$，$b=6$，$n=9$

   (d)$a=4$，$b=4$，$n=15$　　(e)$a=3$，$b=11$，$n=2$　　(f)$a=10$，$b=11$，$n=12$

3. 证明下面关于同余的事实.

   (a)如果 $a\equiv b(\mathrm{mod}\ n)$，那么 $b\equiv a(\mathrm{mod}\ n)$.

   (b)如果 $a\equiv b(\mathrm{mod}\ n)$且 $b\equiv c(\mathrm{mod}\ n)$，那么 $a\equiv c(\mathrm{mod}\ n)$.

   (c)如果 $a\equiv a'(\mathrm{mod}\ n)$且 $b\equiv b'(\mathrm{mod}\ n)$，那么 $a+b\equiv a'+b'(\mathrm{mod}\ n)$.

   (d)如果 $a\equiv a'(\mathrm{mod}\ n)$且 $b\equiv b'(\mathrm{mod}\ n)$，那么 $a\times b\equiv a'\times b'(\mathrm{mod}\ n)$.

4. 假设 $c_1c_2\cdots c_n$ 是 0，1，2，…，$n-1$ 的任意排列. 如下构建矩阵 A. A 的第一行是排列 $c_1c_2\cdots c_n$. A 的每一个后继行都是通过把前一行的各元素加上 1 并做模 n 的加法得到的.

   (a)从排列 32401 开始构造 A.

   (b)证明 A 总是拉丁方.

5. (Williams[1949])设 $n=2m$，且设

$$0\quad 1\quad 2m-1\quad 2\quad 2m-2\quad 3\quad 2m-3\quad \cdots\quad m+1\quad m$$

   是{0，1，2，…，$n-1$}的一个排列. 设 A 是通过练习 4 的方法由这个排列构造的拉丁方.

   (a)对于 $m=2$，构造 A.

   (b)证明对于每一个 $m\geqslant 1$ 的值，A 在下面的意义下是水平完全的(horizontally complete)：只要 $1\leqslant\alpha\leqslant n$，$1\leqslant\beta\leqslant n$，且 $\alpha\neq\beta$，则存在 A 的一行，在这行中 β 紧随 α 之后(这样的拉丁方在农业实验中很重要，在这样的实验中，我们希望最小化邻近土地的处理所带来的相互影响).

   (c)对于任意的 $m\geqslant 1$，(在显而易见的意义下)A 是垂直完全的(vertically complete)吗?

6. 与使用四位数字表示年代的做法不同，有些计算机软件使用从 1970 年 1 月 1 日开始已经过去的秒数表示日期. 而这个秒数用 32 位存储单元以二进制存储. 最终，从 1970 年 1 月开始的秒数会超过 32 位存储单元，造成一个不正确的日期表示. 这一问题称为 Unix 时间问题.

   (a)对于什么样的 n，这一表示法使用模 n 同余?

   (b)Unix 时间问题首次出现在什么时候?

7. 回想一下 9.3.2 节中的值：$p=37$，$q=23$，$r=851$，$s=5$ 和 $t=317$.

   (a)如果这家商店接收到数 $m=852$，确定顾客的信用卡号码.

   (b)对于信用卡号码 123-45-6789，确定 m.

523

   (c)确定另一组可能的 s，t 对. 即确定正整数 s 和 t，使得对于某个正整数 j，有 $st=j(37-1)(23-1)+1$ (请避免 s 或 t 是 1 的平凡情况).

8. 根据 9.3.2 节，假设这家商店选中了素数 $p=37$ 及 $q=41$. 那么 $r=1517$.

   (a)确定正整数 s 和 t，使得对于某个正整数 j，有 $st=j(37-1)(41-1)+1$.(请避免 s 或 t 等于 1 的平凡情况.)

   (b)确定信用卡号是 123-45-6789 的 m.

   (c)如果这家商店接收了数 1163，确定信用卡号码.

9. 本练习给出费马小定理(定理 9.8)的另一个证明.

   (a)证明：如果 p 是素数，那么对于 $1\leqslant i\leqslant p-1$，$p$ 整除二项式系数 $\binom{p}{i}$.

   (b)通过对正整数 x 作数学归纳，使用二项式定理展开 $(x+1)^p$ 证明 $x^p\equiv x(\mathrm{mod}\ p)$.

10. 写出下列域的加法和乘法表.

   (a)GF(5)　　(b)GF(7)　　(c)GF(9)

11. (a)写出 Z_4 上模 4 的加法和乘法的二元运算表.

(b)确定 Z_4 中一个没有乘法逆的元素.

12. 对于集合 Z_{10} 上模 10 的加法和乘法，重复练习 11.
13. 对于下列每个域，确定 8 的加法逆和乘法逆.

(a)GF(11)　　(b)GF(13)　　(c)GF(17)

14. 用 6 取代 8 重复练习 13.
15. 下面的三元组(F，+，×)中，哪些定义域?

(a)F 是正实数，+和×就是普通的加法和乘法.

(b)F 是实数且带有一个额外的元素∞. 运算+和×是通常实数上的加法和乘法，除此之外对于所有的实数 a，我们还有

$$a+\infty=a\times\infty=\infty+a=\infty\times a=\infty+\infty=\infty\times\infty=\infty$$

(c)F 是 Re，$a+b$ 是通常的加法，而对于 F 中的所有 a，b，有 $a\times b=1$.

(d)F 是 Re，对于 F 中的所有 a，b 有 $a+b=a\times b=0$.

16. 考虑模 n 的加法和乘法下的 Z_n，并考虑域的条件.

(a)证明条件 **F1** 成立.

524 (b)证明条件 **F2** 成立.

(c)证明条件 **F3** 成立.

(d)通过证明 0 和 1 分别是加法和乘法的单位元来证明条件 **F4** 成立.

(e)通过证明 $n-a$ 是 a 的加法逆来证明条件 **F5** 成立.

(f)证明条件 **F6** 成立.

(g)如果 n 不是素数，证明条件 **F5**(b)不成立.

(h)如果 n 是素数，证明条件 **F5**(b)成立[提示：利用费马小定理(定理 9.8)得出结论$a^{-1}=a^{n-2}$].

17. 验证表 9.21 定义一个域.
18. 使用 9.3.4 节的方法确定下列各秩的一个拉丁方完全正交族[(c)和(d)是为那些有现代代数学知识的读者所准备的].

(a)5　　(b)7　　(c)8　　(d)9

19. 对于给定的素数 n 和整数 k，$3\leqslant k\leqslant n$，如下构建 $n^2\times k$ 的矩阵 A：对应于每一个满足 $1\leqslant i$，$j\leqslant n$的序对(i，j)构建一行. 如果 $1\leqslant c\leqslant k$，对应于(i，j)的行和列 c 的项是 $i+jc(\text{mod } n)$. 证明 A 是一个正交阵列.

9.4　平衡不完全区组设计

9.4.1　(b，v，r，k，λ)设计

在 9.1 节，我们指出在区组设计中，不是总有可能测试到每一个区组中的每一个处方. 例如，在测试轮胎磨损中，如果存在五个轮胎品牌，那么正如我们所观察的那样，在任意一个区组中只有四种品牌的轮胎得以测试. 因此，有必要使用一个不完全的区组设计. 我们将要研究的基本的不完全区组设计称为平衡不完全区组设计. *平衡区组设计*(balanced block design)是由 $v\geqslant 2$ 个元素的集合 V 和 V 的 $b>0$ 个子集的集合组成的，我们称 V 的元素为*变元*(variety)或*处方*(treatment)，而称 V 的子集为*区组*(block)，且满足下列的条件：

每一个区组正好由相同数量 k 个变元组成，$k>0$；　(9.10)

每一个变元正好在相同数量 r 个区组中出现，$r>0$；　(9.11)

每一对变元正好同时在相同数量 λ 个区组中出现，$\lambda>0$.　(9.12)

满足 $k<v$ 的平衡区组设计称为*平衡不完全区组设计*(balanced incomplete block design)，因为每个区组中的变元数量小于变元的总数. 这样的设计也称为 BIBD、(b，v，r，k，λ)设计((b，

v, r, k, λ)-design)或(b, v, r, k, λ)构形((b, v, r, k, λ)-configuration). BIBD其内在的基本思想是由Yates[1936]引入的. 注意，如果$k=v$，且没有区组有重复变元，那么条件(9.10)、条件(9.11)和条件(9.12)显然成立，且$k=v$，$r=b$和$\lambda=b$. 这就是为什么除特别声明外，我们都假设$k<v$的理由. 525

【例9.10 (7, 7, 3, 3, 1)设计】 如果$b=7$，$v=7$，$r=3$，$k=3$，且$\lambda=1$，那么存在(b, v, r, k, λ)设计. 通过取变元V为集合{1, 2, 3, 4, 5, 6, 7}，且使用下面各区组就给出这个设计：

$$B_1=\{1,2,4\},\quad B_2=\{2,3,5\},\quad B_3=\{3,4,6\},$$
$$B_4=\{4,5,7\},\quad B_5=\{5,6,1\},\quad B_6=\{6,7,2\},\quad B_7=\{7,1,3\}.$$

容易看到，每一个区组正好由3个变元组成，每一个变元正好在3个区组中出现，每一对变元正好同时在一个区组中出现. (例如，3和6正好同时在B_3中出现，而在其他区组中不出现.) ■

【例9.11(4, 4, 3, 3, 2)设计】 如果$b=4$，$v=4$，$r=3$，$k=3$，$\lambda=2$，那么下列集合和区组给出一个(b, v, r, k, λ)设计：

$$V=\{1,2,3,4\},$$

且区组

$$\{1,2,3\},\quad \{2,3,4\},\quad \{3,4,1\},\quad \{4,1,2\}.$$ ■

在9.1节的轮胎磨损例子中，变元是轮胎的品牌，而区组则对应于用于给定测试汽车的轮胎的品牌集合. 条件(9.10)、条件(9.11)、条件(9.12)对应于下面的合理要求：

每一辆汽车使用相同数量k个轮胎品牌；

每一种品牌出现在相同数量r辆汽车上；

每一对品牌在相同汽车上一起测试相同的次数λ.

根据我们对拉丁方正交族的经验可知，不一定对参数b, v, r, k和λ的所有组合都存在(b, v, r, k, λ)设计. 事实上，平衡不完全区组设计的基本组合问题是存在问题：对于什么样的b, v, r, k和λ的组合存在(b, v, r, k, λ)设计？下面我们阐述这个问题. 一般地，描述使得(b, v, r, k, λ)设计存在的参数b, v, r, k, λ的充分必要的完全条件是一个不可解问题. 关于实际实验设计的典型参考书列出了这些参数的一些合理值，即(b, v, r, k, λ)设计存在的那些参数值. 现在，让我们给出使用(b, v, r, k, λ)设计的一些例子. 然后，我们将较详细地研究这一基本的存在问题.

【例9.12 布料磨损测试(再探例9.6)】 假设有如例9.6所描述的Martindale磨损试验机，而且我们希望使用这一台机器比较7种不同类型的布料. 因为在这一台机器的一次运行只能测试4种布料，所以必须使用不完全区组设计. 变元的数量是7，而所有区组的大小是4. Box，Hunter和Hunter[1978]对下面的情形描述了一个(b, v, r, k, λ)设计：在这一设计中有7个区组($b=7$)，每一类型的布料运行$r=4$次，每一对类型的布料在一次运行中一起测试$\lambda=2$次. 如果布料的类型被标上标签1，2，3，4，5，6，7，所使用的区组可以描述如下： 526

$$B_1=\{2,4,6,7\},\quad B_2=\{1,3,6,7\},\quad B_3=\{3,4,5,7\},$$
$$B_4=\{1,2,5,7\},\quad B_5=\{2,3,5,6\},\quad B_6=\{1,4,5,6\},\quad B_7=\{1,2,3,4\}.$$

因为有4个位置放置布料，且因为可以这样选择设计，使得每一种布料在4次运行中被使用，所以也有可能安排在每一个位置每一种类型的布料正好被放置一次. 因此，可以控制机器的位置所带来的差异. 对两种不同的区组偏差源是平衡的这样的不完全设计称为*尤登方*(Youden

square)，这一名字是以这一设计的发明者 W. J. Youden 的名字命名的(参见 Youden[1937])．对此例，尤登方可以如表 9.24 所示的那样进行概括，其中 p，q，r，s 代表 4 个位置，而 i，j 项给出区组 B_j 中变元 i 的位置． ■

表 9.24 磨损测试的尤登方

变元(布料类型)	区组(运行)						
	B_1	B_2	B_3	B_4	B_5	B_6	B_7
1		p		q		r	s
2	p			s	r		q
3		q	p		s		r
4	q		r			s	p
5			s	r	q	p	
6	r	s			q	p	
7	s	r	q	p			

注：i，j 项给出区组 B_j 中变元 i 的位置．

【例 9.13 牛肺结核】 Wadley[1948]把平衡不完全区组设计用于牛肺结核的诊断工作中．这种疾病可以通过在牛的皮肤上注射适当的过敏源并观察所产生的皮肤变厚现象来加以诊断．在一个比较各种过敏源的实验中，对每一种过敏源观察产生 3 毫米厚度时所需的浓度对数(曲线)．可以通过观察四种不同浓度时的厚度并使用插值法来评估这一浓度对数．因此，一种过敏源的每一次测试都需要以不同浓度多次注射这种过敏源．

527

在 Wadley 的实验中，要对 16 种过敏源做比较．因此，$v=16$．在每一头牛身上，有 4 个主要区域，在每一个区域可以进行大约 16 次注射．这表明使用每一个区域作为一个区组．每个区组有 4 种过敏源制剂，每一种过敏源在不同的浓度下使用 4 次，总共进行 16 次注射．因此，$k=4$．存在一种 $k=4$，$v=16$，$b=20$，$r=5$ 的设计．任何典型参考书都有这些信息．因为有 20 个区组，且每一头牛有 4 个区组；这就需要 5 头牛(或者，通过重复实验，需要 5 的整数倍头牛；Wadley 的实验使用了 10 头牛)．如果对于 $k=4$，$v=16$ 没有适当的设计，那么很自然地要考虑是否在每一个区域内要包括 5 种过敏源(每一个区域进行 20 次注射)． ■

【例 9.14 洗涤剂的比较】 在诸如 Pugh[1953]所做的比较家庭所用的洗碗洗涤剂的实验中，要使用下面的过程．为了得到一系列相似的实验单元，把餐厅某道菜的盘子分成几组．每一组盘子用标准温度的水加上一定量的一种洗涤剂来清洗．实验员记录泡沫在减少到薄层之前洗过的盘子数量(的对数)．洗涤剂构成变元．来自餐厅某道菜的每组盘子形成一个实验单元，来自同一道菜但不同组的盘子形成一个区组．一个人洗涤一个区组．一道菜的每一组盘子被指定给一个变元．实验条件在一个区组内尽量不变．不同的区组由不同方式弄肮的盘子及不同人洗涤的盘子组成．

现在，一个区组内的可用盘子数量是有限的．通常只允许四个测试；即限制为 4 个实验单位，因此也限制每一个区组的变元．如果要比较 8 个变元，不能在每一个区组内测试每一个变元．这就需要一个 $v=8$，$k=4$ 的不完全区组设计．对于 $r=7$，$b=14$ 存在这样的设计．

总之，实验员取一组给定菜的盘子，把它们分成 4 组，对每一组使用不同的洗涤剂．这一实验被重复 14 次，每一次实验都使用由 4 种洗涤剂，而这 4 种洗涤剂形成对应区组中的 4 个变元． ■

9.4.2 (b, v, r, k, λ)设计存在的必要条件

我们的第一个定理给出平衡不完全区组设计的参数必须满足的必要条件．

定理 9.11 在一个(b, v, r, k, λ)设计中，有

$$bk = vr \tag{9.13}$$

及

528

$$r(k-1) = \lambda(v-1). \tag{9.14}$$

为了说明这一定理，注意不存在(12，9，4，3，2)设计，因为 $bk=36$，$vr=36$，$r(k-1)=8$，$\lambda(v-1)=16$．因此，尽管(9.13)式成立，但是(9.14)式不成立．如果 $b=12$，$v=9$，$r=4$，$k=3$，$\lambda=1$，那么有 $bk=vr=36$ 及 $r(k-1)=\lambda(v-1)=8$，所以(9.13)式和(9.14)式都

成立. 这说明可能存在(12, 9, 4, 3, 1)设计；但是不能保证这样的设计一定存在[条件(9.13)和条件(9.14)是必要的但不是充分的].

定理 9.11 的证明 乘积 bk 是区组数(b)与每一个区组中的变元数(k)的乘积，因此，如果如下写出各区组，这一乘积给出所列元素的总数：

$$B_1: \cdots \quad B_2: \cdots \cdots \cdots \quad B_b: \cdots.$$

乘积 vr 是变元的数量(v)与每一个变元重复的次数(r)的乘积，因此，这一乘积也给出上面所列元素的总数. 因此 $bk=vr$，即(9.13)式成立.

乘积 $r(k-1)$ 是有变元 i 出现的那些区组的数量(r)与变元 i 出现的每一个区组中其他变元的数量的乘积，因此，这一乘积给出出现于公共区组中的对$\{i, j\}$的数量(对该对的每一次出现计数一次). 乘积 $\lambda(v-1)$ 是每个对$\{i, j\}$在一个区组中出现的次数(λ)与 j 的可能数量 $v-1$ 的乘积，因此，这一乘积给出对$\{i, j\}$在一个公共区组中出现的数量(它每出现一次计数一次). 因此，$r(k-1)=\lambda(v-1)$，因此，(9.14)式成立. ■

推论 9.11.1 *假设在一个有 $v\geqslant 2$ 个变元和 b 个区组的不完全区组设计中，*

1. *每一个区组是由相同的变元数量 k 组成的，且*
2. *每一对变元同时正好在相同数量 λ 个区组中出现，且 $\lambda>0$.*

那么，每一个变元在相同数量 r 个区组中出现，r 由 $\lambda(v-1)/(k-1)$ 给出，且这一区组设计是平衡的.

证明 上面(9.14)式的证明实际上已给出这一推论的证明. 因为假设给定变元 i 在 r_i 个区组中出现. 上面的证明已给出

$$r_i(k-1)=\lambda(v-1).$$

注意，因为 $v\geqslant 2$ 且 $\lambda>0$，所以 k 不能等于 1. 因此，

$$r_i=\frac{\lambda(v-1)}{k-1}.$$

对于每个 i，其结果 r_i 都是相同的. ■ 529

推论 9.11.1 表明，平衡不完全区组设计的定义是冗余的：其定义中的条件之一，即每一个变元在相同数量的区组中出现的条件(9.11)可以从条件(9.10)和条件(9.12)得到.

定理 9.12(费希尔不等式[⊖]) *在(b, v, r, k, λ)设计中，$b\geqslant v$.*

我们将在 9.4.3 节证明这一结果. 为了证明这一结果，需要引入一个很有帮助的概念，这个概念对第 10 章中错误校正的研究也很有用. 这就是区组设计的**关联矩阵**(incidence matrix). 如果一个设计有变元 x_1, x_2, $\cdots$, x_v，区组 B_1, B_2, $\cdots$, B_b，那么 A 是一个由 0 和 1 组成的 $v\times b$ 矩阵，其中，如果 x_i 在 B_j 中则 A 的 i, j 项是 1，否则它是 0(这就是 3.7 节中的点集关联矩阵). 例如，对于例 9.10 中的(b, v, r, k, λ)设计，我们有下面的关联矩阵：

$$A=\begin{array}{c} \\ 1\\2\\3\\4\\5\\6\\7 \end{array}\begin{array}{c} \begin{array}{ccccccc} B_1 & B_2 & B_3 & B_4 & B_5 & B_6 & B_7 \end{array} \\ \begin{bmatrix} 1&0&0&0&1&0&1\\ 1&1&0&0&0&1&0\\ 0&1&1&0&0&0&1\\ 1&0&1&1&0&0&0\\ 0&1&0&1&1&0&0\\ 0&0&1&0&1&1&0\\ 0&0&0&1&0&1&1 \end{bmatrix}\end{array}$$

⊖ 本定理归功于 Fisher[1940].

可以通过给定 0 和 1 的 $v\times b$ 矩阵来定义设计. 对于 $v\geq 2$，任意 $v\times b$ 的 0 和 1 矩阵是 (b, v, r, k, λ) 设计的关联矩阵，其中 $b, v, r, k, \lambda>0$，当且仅当下面条件成立：

$$\text{每一列中 1 的数目相同，都是 } k \text{ 个，且 } k>0; \tag{9.15}$$

$$\text{每一行中 1 的数目相同，都是 } r \text{ 个，且 } r>0; \tag{9.16}$$

$$\text{每对行中同时为 1 的列数相同，都是 } \lambda \text{ 个，} \lambda>0. \tag{9.17}$$

我们已经(在推论 9.11.1 中)看到(9.15)式和(9.17)式蕴涵(9.16)式. (9.17)式的类似结果如下：

$$\text{每对列中同时为 1 的行数相同.} \tag{9.18}$$

练习 48 探讨条件(9.15)～条件(9.18)之间的关系.

9.4.3 费希尔不等式的证明[⊖]

530

为了证明费希尔不等式，我们首先证明关于 (b, v, r, k, λ) 设计的关联矩阵的一个结果.

定理 9.13 如果 A 是 (b, v, r, k, λ) 设计的关联矩阵，那么

$$\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}} = (r-\lambda)\boldsymbol{I} + \lambda\boldsymbol{J}, \tag{9.19}$$

其中 $\boldsymbol{A}^{\mathrm{T}}$ 是 $\boldsymbol{A}$ 的转置矩阵，$\boldsymbol{I}$ 是 $v\times v$ 的单位矩阵，$\boldsymbol{J}$ 是所有项为 1 的 $v\times v$ 矩阵.

证明 设 b_{ij} 是 $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ 的 i, j 项. 那么 b_{ij} 是 $\boldsymbol{A}$ 的第 i 行与 $\boldsymbol{A}$ 的第 j 行的内积(inner product)，即

$$b_{ij} = \sum_{k=1}^{b} a_{ik}a_{jk}.$$

如果 $i=j$，我们看到 $a_{ik}a_{jk}$ 是 1，当且仅当第 i 个变元属于第 k 个区组. 因此，b_{ii} 计数 i 所属的区组的数量，即 r. 如果 $i\neq j$，那么 $a_{ik}a_{jk}$ 是 1 当且仅当第 i 个和第 j 个变元都属于第 k 个区组. 因此，b_{ij} 计数第 i 个和第 j 个变元二者所属的区组的数量，即 λ. 把这些结论翻译成矩阵语言给出(9.19)式. ■

费希尔不等式(定理 9.12)的证明 我们假设 $b<v$，并推导出一个矛盾. 设 $\boldsymbol{A}$ 是关联矩阵. 因为 $b<v$，所以我们可以把 $v-b$ 个 0 列加到 $\boldsymbol{A}$ 上，结果给出一个 $v\times v$ 方阵 $\boldsymbol{B}$. 现在 $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}=\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}$，因为 $\boldsymbol{A}$ 的两行的内积等于 $\boldsymbol{B}$ 的两行的内积. 取行列式，我们得出下面的结论：

$$\det(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}) = \det(\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}) = (\det\boldsymbol{B})(\det\boldsymbol{B}^{\mathrm{T}}).$$

但是，$\det\boldsymbol{B}=0$，因为 $\boldsymbol{B}$ 有 0 列. 因此，$\det(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})=0$. 现在，根据定理 9.13，有

$$\det(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}) = \det\begin{bmatrix} r & \lambda & \lambda & \lambda & \cdots & \lambda \\ \lambda & r & \lambda & \lambda & \cdots & \lambda \\ \lambda & \lambda & r & \lambda & \cdots & \lambda \\ \lambda & \lambda & \lambda & r & \cdots & \lambda \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lambda & \lambda & \lambda & \lambda & \cdots & r \end{bmatrix}. \tag{9.20}$$

从等式(9.20)的右边的矩阵的每一个其他列中减去第一列不改变行列式. 因此

$$\det(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}) = \det\begin{bmatrix} r & \lambda-r & \lambda-r & \lambda-r & \cdots & \lambda-r \\ \lambda & r-\lambda & 0 & 0 & \cdots & 0 \\ \lambda & 0 & r-\lambda & 0 & \cdots & 0 \\ \lambda & 0 & 0 & r-\lambda & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lambda & 0 & 0 & 0 & \cdots & 0 \\ \lambda & 0 & 0 & 0 & \cdots & r-\lambda \end{bmatrix}. \tag{9.21}$$

531

⊖ 本小节可以跳过.

在(9.21)式的右边的矩阵的第一行加上所有其他行不改变这个行列式. 因此，有

$$\det(\boldsymbol{AA}^{\mathrm{T}})=\det\begin{bmatrix} r+(v-1)\lambda & 0 & 0 & 0 & \cdots & 0 \\ \lambda & r-\lambda & 0 & 0 & \cdots & 0 \\ \lambda & 0 & r-\lambda & 0 & \cdots & 0 \\ \lambda & 0 & 0 & r-\lambda & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lambda & 0 & 0 & 0 & \cdots & 0 \\ \lambda & 0 & 0 & 0 & \cdots & r-\lambda \end{bmatrix}. \tag{9.22}$$

因为(9.22)式的右边矩阵的对角线的上方都是0，所以它的行列式等于对角线上元素的积，所以有下面的等式：

$$\det(\boldsymbol{AA}^{\mathrm{T}})=[r+(v-1)\lambda](r-\lambda)^{v-1}.$$

因为我们已得出结论 $\det(\boldsymbol{AA}^{\mathrm{T}})=0$，所以我们有

$$[r+(v-1)\lambda](r-\lambda)^{v-1}=0. \tag{9.23}$$

但是因为 r，v 和 λ 都假设是正的，所以有

$$[r+(v-1)\lambda]>0.$$

同样，根据定理9.11的等式(9.14)，因为 $k<v$，所以有 $r>\lambda$. 因此有

$$(r-\lambda)^{v-1}>0.$$

我们得出结论：(9.23)式的左边是正的，而这是一个矛盾. ■

9.4.4 可分解设计

我们说一个(b, v, r, k, λ)设计是可分解的，如果可以把其中的区组划分成称为平行类(parallel class)的若干组，使得每一个平行类中的区组依次划分变元的集合. 例如，表9.25给出由4个平行类组成的(12, 9, 4, 3, 1)设计. 注意，在每一个平行类中，3个区组是不相交的，且它们的并集是 $V=\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

表9.25 带有平行类 C_1，C_2，C_3，C_4 的(12, 9, 4, 3, 1)设计

C_1:{1,2,3},{4,5,6},{7,8,9}
C_2:{1,4,7},{2,5,8},{3,6,9}
C_3:{1,5,9},{2,6,7},{3,4,8}
C_4:{1,6,8},{2,4,9},{3,5,7}

【例9.15 匿名门限方案和秘密分享】⊖ 在例9.9中，我们介绍了秘密分享的(q, p)门限方案. 在一个匿名(q, p)门限方案(anonymous(q, p)-threshold scheme)中，p个人接收到信息的p个不同部分片段，且密钥可以从任意q个部分片段计算可得，而不必知道谁持有哪一片. 在例9.9中，我们由正交阵列构建的门限方案不是匿名的(参见练习13). 我们将看到可分解(b, v, r, k, λ)设计是如何帮助我们寻找匿名(q, p)门限方案，特别是匿名$(2, p)$门限方案的. [532]

假设我们有一个可分解(b, v, r, k, λ)设计，且$\lambda=1$，每个区组里有$k=p$个变元. 设C_1，C_2，…，C_r是平行类. 因此，每个平行类有v/p个区组. 每一个变元在每一个平行类中出现一次，所以平行类的数量可由等式(9.14)计算，为

$$r=\frac{\lambda(v-1)}{p-1}=\frac{(v-1)}{p-1}.$$

假设这个组的所有p个人都知道这些平行类. 让我们取可能钥匙的集合K为$\{1, 2, \cdots, r\}$，且部分信息片段的集合P为V，即变元的集合. 假设领导人想要分享密钥$k\in K$. 那么他从平行类C_k中随机选取一个区组，并把这个区组中的p个片段的部分信息给这p个人，每一个参

⊖ 本例来自于Stinson[2003].

与者一条信息. 注意，现在任意两个人能够确定这个密钥 k. 因为，如果 p_u 是给人 u 的部分信息，而且 p_v 是给人 v 的部分信息，那么，因为 $\lambda=1$，所以这一设计中存在唯一一个包含变元 p_u 和 p_v 的区组. 人 u 和人 v 能够找到这个区组，并知道如果包含 p_u 和 p_v 的区组是在平行类 C_k 中，则这个密钥是 k. 注意，这是匿名的，因为哪个人持有哪个钥匙无关紧要，重要的是这两个人持有哪些钥匙.

任何持有部分信息 p_u 的人 u 能确定出这一密钥的概率是多少吗？每一个平行类正好有一个包含 p_u 的区组. 因此，正确地猜测出这位领导人心中的密钥/平行类的概率是 $1/r$，其中 r 是平行类的数量. 这个概率也正好等于没有给出部分信息时正确猜测的概率. 如果 r 非常大，那么我们就有一个非常安全的匿名(2，p)门限方案.

例如，表 9.25 的可分解(12，9，4，3，1)设计可以用于构建匿名(2，3)门限方案. 如果领导者想要分享密钥 3，那么他在 C_3 中随机选出一个区组，比如说是{3，4，8}. 这个部分信息分配给这个组中的 3 个人. 例如，得到 3 和 8 的两个人知道领导人想要的唯一区组是{3，4，8}，它在 C_3 中，所以知道密钥是 3. ■

9.4.5 施泰纳三元系统

迄今为止，我们的结果给出(b，v，r，k，λ)设计存在的必要条件，但是这些结果并没有给出其存在的充分条件，或者给出确定它们的构造过程. 我们将描述若干这样的过程. 我们从考虑几个特殊的(b，v，r，k，λ)设计开始.

533

特别地，假设 $k=2$ 及 $\lambda=1$. 在这种情况下，每个区组由两个变元组成，且每一对变元正好在一个区组中出现. 等式(9.14)表明 $r=v-1$，所以(9.13)式表明

$$2b=v(v-1),$$

或者

$$b=\frac{v(v-1)}{2}.$$

现在

$$\frac{v(v-1)}{2}=\binom{v}{2}.$$

这是 v 集合的 2 子集的数量. 因此，区组的数量是变元集合的 2 子集的数量. 例如，如果 $v=3$，那么 $V=\{1，2，3\}$的设计中的区组是下面的子集：

$$\{1,2\},\{1,3\},\{2,3\}.$$

本小节主要研究(b，v，r，k，λ)设计的另一种特殊情况，其中 $k=3$ 且 $\lambda=1$. 这样的设计是一个三元组的集合，其中每一对变元正好在一个三元组中出现. 这些设计称为*施泰纳三元系统*(Steiner triple system). 有些作者定义施泰纳三元系统为这样的区组设计，在该设计中，区组是集合 V 的三元组，且每一对变元正好在一个三元组中出现. 这一定义包含 $k=v$ 的完全区组设计. 这是一个平凡设计，其中 $V=\{1，2，3\}$，且只有一个区组{1，2，3}. 对本小节来说，我们的施泰纳三元系统包含这样的设计. $v=7$ 是施泰纳三元系统的一个更有趣的例子. 例 9.10 中的(7，7，3，3，1)设计就是这样的例子.

我们将讨论施泰纳三元系统的存在问题. 注意，在施泰纳三元系统中，(9.14)式表明

$$r(2)=v-1, \tag{9.24}$$

所以

$$r=\frac{v-1}{2}. \tag{9.25}$$

等式(9.13)表明

$$3b=\frac{v(v-1)}{2},$$

所以

$$b=\frac{v(v-1)}{6}. \tag{9.26}$$

等式(9.25)表明 $v-1$ 是偶数，所以 v 是奇数. 而且，$v\geqslant 2$ 表明 v 至少等于 3. 等式(9.26)表明 $v(v-1)=6b$,所以 $v(v-1)$是 6 的倍数. 这是必要条件. 我们从列出满足这两个必要条件的 v 值开始：v 是奇数且至少为 3，$v(v-1)$是 6 的倍数. 如果 $v=3$ 那么 $v(v-1)=6$，所以有可能存在 $v=3$的施泰纳三元系统，即必要条件成立. 然而 $v=5$ 时 $v(v-1)=20$，这不能被 6 整除，所以不存在 $v=5$ 的施泰纳三元系统. 更一般地，对于 $v=3$，7，9，13，15，19，21，…，即对 $v=6n+1$ 或 $6n+3$，$n\geqslant 1$ 和 $v=3$ 的情况有可能存在施泰纳三元系统. 事实上，对于所有这些 v 的值，确实存在这样的系统. [534]

定理 9.14(Kirkman[1847]) *存在 v 个变元的施泰纳三元系统，当且仅当 $v=3$，或者 $v=6n+1$ 或 $v=6n+3$，且 $n\geqslant 1$.*

我们已经证明了定理 9.14 的必要条件. 下面我们不给出充分条件的证明，而是证明一个更简单的定理. 这个定理给出了其中若干个施泰纳三元系统的存在性：例如，有 3・3=9 个变元，3・7=21 个变元，7・7=49 个变元，9・7=63 个变元等的施泰纳三元组. 对于充分性的若干证明，参见 Lindner 和 Rodger[1997].

定理 9.15 *如果存在 v_1 个变元和 v_2 个变元的施泰纳三元系统 S_1 和 S_2，那么存在 v_1v_2 个变元的施泰纳三元系统 S.*

证明[⊖] 这一定理的证明给出了已知两个施泰纳三元系统 S_1 和 S_2，构造一个新的施泰纳三元系统 S 的方法. 假设 S_1 的变元是 a_1，a_2，…，a_{v1}而 S_2 的变元是 b_1，b_2，…，b_{v2}. 设 S 是由 v_1v_2 个元素 C_{ij}组成的，其中 $i=1$，2，…，v_1，$j=1$，2，…，v_2. 三元组$\{c_{ir}, c_{js}, c_{kt}\}$在 S 中当且仅当下面的条件之一成立：

(1)$r=s=t$ 且$\{a_i, a_j, a_k\}\in S_1$，

(2)$i=j=k$ 且$\{b_r, b_s, b_t\}\in S_2$，

或者

(3)$\{a_i, a_j, a_k\}\in S_1$ 且$\{b_r, b_s, b_t\}\in S_2$.

于是，容易证明 S 形成一个施泰纳三元系统. ■

让我们说明一下定理 9.15 的证明中给出的构造法. 假设 $v_1=v_2=3$，S_1 有一个三元组$\{a_1, a_2, a_3\}$而 S_2 有一个三元组$\{b_1, b_2, b_3\}$. 于是 S 有如表 9.26 所示的三元组，而且形成 9 个变元和 12 个区组的施泰纳三元系统.

表 9.26 当 S_1 只有三元组$\{a_1, a_2, a_3\}$且 S_2 只有三元组$\{b_1, b_2, b_3\}$时，根据 S_1 和 S_2，$v_1v_2=9$ 变元的施泰纳三元系统 S 的构造法

定理 9.15 的证明中的条件(1)	$r=s=t=1$ $\{c_{11}, c_{21}, c_{31}\}$	$r=s=t=2$ $\{c_{12}, c_{22}, c_{32}\}$	$r=s=t=3$ $\{c_{13}, c_{23}, c_{33}\}$
定理 9.15 的证明中的条件(2)	$i=j=k=1$ $\{c_{11}, c_{12}, c_{13}\}$	$i=j=k=2$ $\{c_{21}, c_{22}, c_{23}\}$	$i=j=k=3$ $\{c_{31}, c_{32}, c_{33}\}$
定理 9.15 的证明中的条件(3)	$\{c_{11}, c_{22}, c_{33}\}$ $\{c_{11}, c_{23}, c_{32}\}$	$\{c_{12}, c_{23}, c_{31}\}$ $\{c_{12}, c_{21}, c_{33}\}$	$\{c_{13}, c_{21}, c_{32}\}$ $\{c_{13}, c_{22}, c_{31}\}$

⊖ 本定理的证明和说明可以跳过.

当一个实验设计是 v 个变元的一个施泰纳三元系统时，如果 $v=3$，7，9，那么特定的设计选择是简单的，因为在这些情况下(在重标签变元的意义下)只存在一个 v 个变元的施泰纳三元系统．然而，对于 $v=13$，实质上存在两个不同的施泰纳三元系统，对于 $v=15$，存在 80 个不同的系统，而对于 $v=19$，存在 11 084 874 829 个不同的系统(Kaski 和 Östergard[2004])．如果需要 13，15 或者 19 变元的施泰纳三元系统，大概可以随机地从中选择一个．一般地，对于 $v>19$，当存在施泰纳三元系统时，这样的不同施泰纳三元系统的数量是未知的．

535

9.4.6　对称平衡不完全区组设计

一个平衡不完全区组设计或(b，v，r，k，λ)设计称为是**对称的**(symmetric)，如果 $b=v$(区组的数量等于变元的数量)而且 $r=k$(一个变元出现的次数等于一个区组中变元的数量)．对称 BIBD 有时称为(v，k，λ)**设计**(design)或(v，k，λ)**构形**(configuration)．根据定理 9.11 的等式(9.13)，有

$$b = v \quad \text{当且仅当} \quad k = r.$$

因此，定义中的这两个条件是冗余的．例 9.11 是对称 BIBD 的例子：我们有 $b=v=4$ 和 $r=k=3$．例 9.10 也是对称 BIBD 的例子：我们有 $b=v=7$ 和 $r=k=3$．表 9.26 的施泰纳三元系统是 BIBD 的例子，但它不是对称的．

定理 9.16(Bruck-Ryser-Chowla 定理㊀)　*下面的条件是(v，k，λ)设计存在的必要条件：*

1. 如果 v 是偶数，那么 $k-\lambda$ 是一个整数的平方．

2. 如果 v 是奇数，那么下面的方程有不都为 0 的整数解 x，y，z：

$$x^2 = (k-\lambda)y^2 + (-1)^{(v-1)/2}\lambda z^2. \tag{9.27}$$

我们省略定理 9.16 的证明，对于这一证明，请参见 Ryser[1963]和 Hall[1967]．为了说明这一定理，假设 $v=16$，$k=6$ 及 $\lambda=2$．那么 v 是偶数，且 $k-\lambda=4$ 是一个平方数，所以条件 1 表明(16，6，2)设计可能存在．然而条件 1 还表明(22，7，2)设计不可能存在，因为 $k-\lambda=5$ 不是平方数．假设 $v=111$，$k=11$，和 $\lambda=1$．那么 v 是奇数，且(9.27)式变成

$$x^2 = 10y^2 - z^2.$$

536

这一方程有解 $x=y=1$，$z=3$．因此(111，11，1)设计可能存在．

定理 9.16 中给出的对称 BIBD 存在的条件不是充分的．既使我们已证明(111，11，1)设计可能存在，但 Lam，Thiel 和 Swiercz[1989]证明这一设计不存在．下面的定理给出若干特殊的充分条件，其证明将在 10.5.2 节给出．㊁

定理 9.17　*对于任意大的值 m，特别地对于 $m=2k$，$k\geq 1$，存在($4m-1$，$2m-1$，$m-1$)设计．*

($4m-1$，$2m-1$，$m-1$)设计称为**维数 m 的阿达马设计**(Hadamard design of dimension m)．$m=2$ 的情况给出(7，3，1)设计，这就是例 9.10 中给出的例子．根据定理 9.16，对于所有的 $m\geq 2$，维数 m 的阿达马设计可能存在．因为 $v=4m-1$ 是奇数，而且(9.27)式变成

$$x^2 = my^2 - (m-1)z^2,$$

这个方程有一组解 $x=y=z=1$．阿达马设计在 10.5 节所述的错误校正码理论中非常重要．

下面是给出对称 BIBD 存在的另一个充分条件的定理，它的证明将在 9.5.2 节我们研究射影平面时给出．

㊀ Bruck 和 Ryser[1949]证明了 $\lambda=1$ 时的定理，Chowla 和 Ryser[1950]证明了其一般形式．

㊁ 读者也可以现在阅读 10.5.2 节的内容．

定理 9.18 如果 $m\geqslant1$ 是一个素数的幂，那么存在$(m^2+m+1, m+1, 1)$设计.

为了说明这一定理，注意取 $m=1$ 给出一个$(3, 2, 1)$设计. 在 9.4.5 节的开始我们已看到这样的设计. 取 $m=2$ 给出一个$(7, 3, 1)$设计. 我们在例 9.10 已经看到这样的设计. 取 $m=3$ 给出一个$(13, 4, 1)$设计，这是一个新设计.

构造对称 BIBD 的第三种方法是使用差集. 这一方法将在练习 31 和练习 32 中叙述.

9.4.7 从已有的设计构建新的(b, v, r, k, λ)设计

定理 9.15 给出从已知的设计构建新(b, v, r, k, λ)设计的方法. 这里我们将给出其他的方法. 从一个设计得到另外一个设计的最平凡的方法是简单地重复区组. 如果我们在(b, v, r, k, λ)设计中，每个区组取 p 次拷贝，那么我们得到一个$(pb, v, pr, k, p\lambda)$设计. 例如，根据例 9.11 的$(4, 4, 3, 3, 2)$设计，通过重复其中每一个区组两次，我们得到一个$(8, 4, 6, 3, 4)$设计. 为了描述从一个旧设计得到新设计的更多有趣的方法，我们需要一个预备结果.

定理 9.19 在(v, k, λ)设计中，任意两个区组都正好有 λ 个公共元素.

证明 见练习 48. ■ 537

对于集合 U 和 V，我们用 $U-V$ 表示集合 $U\cap V^c$.

定理 9.20 假设 $B_1, B_2, \cdots, B_v$ 是一个(v, k, λ)设计的区组，且 $V=\{x_1, x_2, \cdots, x_v\}$ 是变元的集合. 那么对于任意 i，

$$B_1-B_i, B_2-B_i, \cdots, B_{i-1}-B_i, B_{i+1}-B_i, \cdots, B_v-B_i$$

是变元集合 $V-B_i$ 上的$(v-1, v-k, k, k-\lambda, \lambda)$设计的区组.

证明 显然存在 $v-1$ 个区组和 $v-k$ 个变元. 根据定理 9.19，每一个区组 B_j-B_i 有 $k-\lambda$ 个元素. $V-B_i$ 中的每一个变元在原来设计的 k 个区组中出现，因此，也在新设计的 k 个区组中出现. 类似地，$V-B_i$ 中的每一对变元同时在原来设计的 λ 个区组中出现，因此也同时在新设计的 λ 个区组中出现. ■

为了说明这一构造法，假设我们开始于例 9.10 的$(7, 3, 1)$设计，且设 $B_i=\{3, 4, 6\}$. 这时，下面的区组形成变元集合$\{1, 2, 5, 7\}$上的$(6, 4, 3, 2, 1)$设计：

$$\{1,2\},\{2,5\},\{5,7\},\{1,5\},\{2,7\},\{1,7\}.$$

定理 9.21 假设 $B_1, B_2, \cdots, B_v$ 是(v, k, λ)设计的区组，且 $V=\{x_1, x_2, \cdots, x_v\}$是变元集合. 于是对于任意的 i，

$$B_1\cap B_i, B_2\cap B_i, \cdots, B_{i-1}\cap B_i, B_{i+1}\cap B_i, \cdots, B_v\cap B_i$$

是变元集合 B_i 上的$(v-1, k, k-1, \lambda, \lambda-1)$设计的区组.

证明 显然存在 $v-1$ 个区组和 k 个变元. 根据定理 9.19，每个区组 $B_j\cap B_i$ 有 λ 个元素. 另外，B_i 中的给定变元在原来设计的如下区组中出现：

$$B_{j_1}, B_{j_2}, \cdots, B_{j_{k-1}}, B_i.$$

于是，这个元素也在新设计的如下 $k-1$ 个区组中出现：

$$B_{j_1}\cap B_i, B_{j_2}\cap B_i, \cdots, B_{j_{k-1}}\cap B_i.$$

另外，B_i 中任意一对变元同时在原来设计的如下 λ 个区组中出现：

$$B_{j_1}, B_{j_2}, \cdots, B_{j_{\lambda-1}}, B_i,$$

因此任意一对变元也同时在新设计的如下 $\lambda-1$ 个区组中出现：

$$B_{j_1}\cap B_i, B_{j_2}\cap B_i, \cdots, B_{j_{\lambda-1}}\cap B_i.$$ ■

为了说明此定理，我们注意到，根据定理 9.17，存在$(15, 7, 3)$设计. 因此，定理 9.21 表明，存在$(14, 7, 6, 3, 2)$设计. 为了给出这样的设计，我们注意到，表 9.27 中的区组定义$\{1,$

2，…，15}上的一个(15，7，3)设计．我们将在10.5.2节中给出如何构建这一设计．取B_i={1，2，3，8，9，10，11}，我们得到表9.28中变元集合B_i上的一个(14，7，6，3，2)设计．注意这一设

538

计有重复的区组．如果我们对这些区组的每一个只取一次拷贝，那么我们得到(7，3，1)设计．

表9.27 变元集合V={1，2，…，15}上的(15，7，3)设计的区组

{2，4，6，8，10，12，14}，	{1，4，5，8，9，12，13}，	{3，4，7，8，11，12，15}，
{1，2，3，8，9，10，11}，	{2，5，7，8，10，13，15}，	{1，6，7，8，9，14，15}，
{3，5，6，8，11，13，14}，	{1，2，3，4，5，6，7}，	{2，4，6，9，11，13，15}，
{1，4，5，10，11，14，15}，	{3，4，7，9，10，13，14}，	{1，2，3，12，13，14，15}
{2，5，7，9，11，12，14}，	{1，6，7，10，11，12，13}，	{3，5，6，9，10，12，15}

表9.28 通过在表9.27中插入区组B_i={1，2，3，8，9，10，11}而得到的(14，7，6，3，2)设计的各区组

{2，8，10}，{1，8，9}，{3，8，11}，{2，8，10}，{1，8，9}，{3，8，11}，{1，2，3}
{2，9，11}，{1，10，11}，{3，9，10}，{1，2，3}，{2，9，11}，{1，10，11}，{3，9，10}

9.4.8 组群测试及其应用

假设把一个大族U的项划分成两类，正的和负的．我们希望确定正项目，但是检查U中的每一个项却又过分昂贵．然而，我们可以把这些项分组成U的子集，并检测一个子集是否至少含有一个正项．我们想通过几组测试确定所有正项．如果我们需要确定所有要进行测试的组群，然后，进行组群测试且不能利用这些测试结果选取要测试的新组群，那么我们讨论的是*不适应组群测试*(nonadaptive group testing)．否则我们讨论的是*适应组群测试*(adative group testing)．组群测试理论深受组合数学方法的影响，特别地，深受组合设计方法的影响．

这里是一些使用组群测试的例子．

(1)**次品鉴别**．在给定产品的销售之前我们希望挑出次品(这里，次品是"正"项)．

(2)**疾病筛查**．我们希望确定一组人中哪些人有特定疾病．这是由Dorfma[1943]在研究组群测试的研究中提出的问题(这一问题与上百万应征入伍者的梅毒疹检查相关)．这一课题对于大规模的HIV筛查非常重要(这里，有这种疾病的人是"正"项)．

(3)**基因组图谱测绘**．我们有一个很长的分子序列S，例如，DNA．我们要建立称为*克隆*(clone)的子串库．我们测试一个特殊的序列，即*试样*(probe)是否出现在S中，其方法是检看这个试样出现在哪些克隆中．我们通过把这些克隆合并成若干组群来进行这一工作，因为一个克隆库有成千上万个克隆．

539

(4)**卫星通信**．很多地面站都是卫星通信线路的潜在用户．在调度卫星线路中对时隙的需求时，我们不与所有地面站逐个联系，而是与多组地面站联系来检查是否有一组中的某个站希望预定卫星时间．

(5)**科学或工业实验**．我们希望确定大量的可能变元中哪些是重要的．如果我们能够假设一个重要变元的影响足够强以至于不能被其他变元掩盖，我们可以首先研究组群里的重要变元以确定重要变元．

关于组群测试的诸多应用、组群测试的相关理论以及参考文献，可以在Du和Hwang[2000]所写的书中找到．也可参见Colbourn，Dinitz和Stinson[1999]．

本节阐述不适应组群测试与(b, v, r, k, λ)设计之间的关系．*不适应组群测试算法*(nonadaptive group testing algorithm)，或NAGTA从有u个元素的一组U开始．这一算法试图

确定 U 中属于正项的某个子集的那些项. 该算法使用由 U 的 g 个子集组成的集合 G, 这个集合称为组群(group). 我们排序 G 中的子集. 组群测试的数据可以用 0 和 1 的向量形式记录, 这个向量的第 i 项是 1 当且仅当第 i 个组群测试为正. 对于 G 中的每一个组群 $X\subseteq U$ 和每一个子集 $P\subseteq U$, 设当 $X\cap P\neq\varnothing$ 时 $f_X(P)$ 给出结果 1, 否则给出结果 0. 对应于 G 中所有组群 X 的位串 $(f_{X_1}(P), f_{X_2}(P), \cdots, f_{X_g}(P))$ 记作 $f_G(P)$. 假设我们可以确定组群 G, 使得对于大小至多是 t 的 U 的所有子集 $Q\neq P$, 有 $f_G(Q)\neq f_G(P)$. 如果是这种情况, 那么, 只要正项的数目至多是 t, 通过验证哪个子集 P 的向量 $f_G(P)$ 与观察向量 v 匹配, 并根据 G 组群测试, 我们可以知道在 U 中哪些项正好是正的. 作为例子, 假设 $U=\{1, 2, 3, 4\}$, 且 G 是由下面的有序序列给出的:

$$G=(\{1,2\},\{1,3\},\{1,4\},\{2,3\},\{2,4\},\{3,4\}),\qquad(9.28)$$

那么

$$f_G(\{2\})=(1,0,0,1,1,0),$$

且

$$f_G(\{1,2\})=(1,1,1,1,1,0).$$

在表 9.29 中, 向量 $f_G(P)$ 是由所有满足 $|P|\leqslant 2$ 的 P 给出的. 所以 $f_G(\{2\})\neq f_G(\{1, 2\})$, 且显然, 对于所有的 $Q\neq P$ 且 $|P|\leqslant 2$, $|Q|\leqslant 2$, 有 $f_G(Q)\neq f_G(P)$. 因此, G 中的 6 个组群测试的结果唯一确定正项子集 P. 又假设 $U=\{1, 2, 3, 4\}$. 下面集合

$$G'=(\{1,2\},\{1,3\},\{1,4\})$$

表 9.29　对于 $U=\{1, 2, 3, 4\}$ 和(9.28)中的 G, 满足 $|P|\leqslant 2$ 的所有 P 的 $f_G(P)$

P	$f_G(P)$
{1}	(1, 1, 1, 0, 0, 0)
{2}	(1, 0, 0, 1, 1, 0)
{3}	(0, 1, 0, 1, 0, 1)
{4}	(0, 0, 1, 0, 1, 1)
{1, 2}	(1, 1, 1, 1, 1, 0)
{1, 3}	(1, 1, 1, 1, 0, 1)
{1, 4}	(1, 1, 1, 0, 1, 1)
{2, 3}	(1, 1, 0, 1, 1, 1)
{2, 4}	(1, 0, 1, 1, 1, 1)
{3, 4}	(0, 1, 1, 1, 1, 1)

540

有 $f_G(\{1\})=f'_G(\{1, 2\})=(1, 1, 1)$, 但是对于 $i\neq j$, 有 $f'_G(\{i\})\neq f'_G(\{j\})$ 我们说 G 定义一个有门限值 t 的成功 NAGTA(successful NAGTA with threshold), 如果对于所有至多有 t 个元素的 P 和 Q(及 $P\neq Q$)有 $f_G(Q)\neq f_G(P)$. 因此, G' 是有门限值 1 的成功 NAGTA, 但不是有极限值 2 的成功 NAGTA. 换句话说, 如果正好存在一个正项, 那么组群 G' 的集合可以用于确定正项, 但存在两个或更多正项时则不能确定正项.

现在, 我们给出当有一个大小为 u 的族 U[⊖] 时, 如何构建有给定门限值 t 的成功 ANGTA. 开始于一个(b, v, r, k, λ)设计, 其中 $b=u$, $k=t+1$, $\lambda=1$. 设 b 个区组对应于族 U 的元素. 设 $V=\{1, 2, \cdots, v\}$ 是变元集合, 并把 G 中的第 i 个组群 X_i 定义为包含有变元 i 的区组的集合. 于是:

- U 中的每一个元素(设计的区组)正好在 k 个组群中.
- 每一个组群中正好有 r 个项来自于 U.
- U 中的每一对不同项至多包含于一个组群中(因为 $\lambda=1$).

读者也许发现, 把对(U, G)看成由(b, v, r, k, λ)设计的关联矩阵的转置所定义是有益的[⊜].

我们将证明 G 有门限值 $t=k-1$. 首先, 为了说明这一构造法, 假设我们开始于例 9.10 的 $(7, 7, 3, 3, 1)$设计. 这时, 族 U 是$\{B_1, B_2, B_3, B_4, B_5, B_6, B_7\}$, 而各组群由下面集合给出:

$$X_1=\{B_1,B_5,B_7\},X_2=\{B_1,B_2,B_6\},X_3=\{B_2,B_3,B_7\},X_4=\{B_1,B_3,B_4\},$$

⊖ 这一构造法源自 Stinson[2003].

⊜ 关联矩阵是已知 BIBD 的关联矩阵的转置的设计称为该设计的对偶(dual).

541

$$X_5=\{B_2,B_4,B_5\},X_6=\{B_3,B_5,B_6\},X_7=\{B_4,B_6,B_7\}.$$

现在，假设我们知道 $f_G(P)=(1,1,1,1,0,1,0)$. 读者可以验证 $f_G(\{B_1,B_3\})=(1,1,1,1,0,1,0)$. 因为我们有门限值 $k-1=2$，所以我们知道 P 一定是 $\{B_1,B_3\}$.

我们通过证明 G 的门限值是 $k-1$ 就可得出结论. 假设 $|P|\leqslant k-1$，$|Q|\leqslant k-1$，且 $P\neq Q$. 我们证明对于某个 j 有 $f_{X_j}(P)\neq f_{X_j}(Q)$. 不失一般性，在 Q 中存在一个不在 P 中的区组 B. 现在，B 正好在 k 个组群 X_{j_1}，X_{j_2}，…，X_{j_k} 中. 如果对于所有的 $i=1,2,\cdots,k$ 有 $X_{j_i}\cap P\neq\varnothing$，那么对于每一个 i，存在区组 $B(i)\in X_{j_i}\cap P$. 现在，如果 $i\neq i'$ 则有 $B(i)\neq B(i')$，因为 B 和 $B(i)$ 至多在一个组群中. 但是，P 至少有 k 个元素，即 $B(1)$，$B(2)$，…，$B(k)$，这与 $|P|\leqslant k-1$ 矛盾. 因此，对于某个 i，且 $1\leqslant i\leqslant k$，有 $X_{j_i}\cap P=\varnothing$. 因此，$f_{X_{j_i}}(P)=0$. 然而，$B$ 在 Q 和 X_{j_i} 中，所以 $f_{X_{j_i}}(Q)=1$.

9.4.9 施泰纳系统和国家彩票[⊖]

英国的国家彩票包括购买从整数 1 到 49 形成的 6 位数彩票. 一周两次随机抽取 6 个“获胜”数字. 带有三个或更多获胜数字的彩票至少可以赢得 10 英镑. 我们的问题是为了保证有一张获奖彩票，必须购买的最少彩票数是多少？为了论述这一问题，我们考虑称为施泰纳系统的区组设计的变形.

施泰纳系统(Steiner system) $S(t,k,v)$ 是由 v 个元素的集合 V 和大小为 k 称为区组的 V 的子集的集合组成的，使得 V 的任意 t 个元素都正好在一个区组中出现. 施泰纳三元系统(9.4.5 节)是 $S(2,3,v)$ 施泰纳系统.

对于很多 t，k 和 v，不存在施泰纳系统. 事实上，对于 $t\geqslant 6$，我们不知道是否存在这样的系统.

定理 9.22 *如果 $S(t,k,v)$ 施泰纳系统存在，那么 $\binom{k}{t}$ 整除 $\binom{v}{t}$ 且区组的数量由 $\binom{v}{t}\Big/\binom{k}{t}$ 给出.*

证明 考虑集合 V 上的 $S(t,k,v)$ 施泰纳系统. V 中存在 $\binom{v}{t}$ 个 t 元素的子集. 每一个元素正好在 V 的一个区组中出现. 而且，作为 k 子集，每个区组包含 V 的 $\binom{k}{t}$ 个 t 子集. 因此，

$$\binom{v}{t}\Big/\binom{k}{t}=S(t,k,v)\text{ 的区组的数量.}$$

如果上面的数不是一个整数，那么 $S(t,k,v)$ 不存在. ∎

$S(3,6,49)$ 施泰纳系统可以带来一种方法，它保证每一个可能的三元组正好在我们的一张彩票上出现.（为什么？）可是，不存在 $S(3,6,49)$ 施泰纳系统，因为

542

$$\binom{49}{3}\Big/\binom{6}{3}=18\,424/20=921.2.$$

修改这一想法(可能一些三元组在一个以上的区组中出现)的方法至少需要 922 个区组或 922 张票(921.2 向上四舍五入). 我们给出一个更有效的方法.

如果彩票只使用 1～26 之间的数，且从中抽取 6 个获胜数字，获奖票必须至少有其中的三个数字时情况如何呢？存在 $S(3,6,26)$ 施泰纳系统(Chen[1972]). 根据定理 9.22，它有 $\binom{26}{3}\Big/\binom{6}{3}=130$ 个区组，所以存在一种每购买 130 张彩票就可以保证有一张获奖彩票的方法.

⊖ 本小节基于 Brinkman，Hodgkinson 和 Humphreys[2001].

现在，我们要说明如何修改这一想法，使得在使用1～49数字的英国国家彩票中，如果一个人购买260张彩票，就保证其中有一张获奖彩票．注意，任意6个获奖数字必须或者包含三个偶数，或者必须包含三个奇数．因此，这一想法是购买足够多的彩票以保证每一个偶数的三元组和每一个奇数的三元组出现．如果$V=\{2, 4, \cdots, 48\}\cup\{1, 3\}$，那么$|V|=26$．使用$V$上的一个$S(3, 6, 26)$施泰纳系统，我们找到130张彩票，这130张彩票保证每一个从2到48的偶数三元组在其中的一张彩票上出现．类似地，使用集合$V=\{1, 3, \cdots, 49\}\cup\{2\}$上的一个$S(3, 6, 26)$施泰纳系统，我们找到另外130张彩票，这130张彩票保证从1～49的每一个奇数三元组出现在其中的一张彩票上．因此，260张彩票就足以保证其中有一张彩票，有三个数字在从1～49所选出的6位数字中．关于概括为所谓的“Lotto设计”的最新论文，参见Li和van Rees[2002]．

本节练习

1. 对于下列每一个区组设计，确定该设计是否是BIBD设计，如果是，确定它的参数b，v，r，k和λ．

   (a)变量：$\{1, 2, 3, 4\}$

   　区组：$\{1, 2\}$，$\{1, 3\}$，$\{2, 4\}$，$\{1, 2, 3\}$，$\{2, 3, 4\}$

   (b)变量：$\{1, 2, 3, 4, 5\}$

   　区组：$\{1, 2, 3\}$，$\{2, 3, 4\}$，$\{3, 4, 5\}$，$\{1, 4, 5\}$，$\{1, 2, 5\}$

   (c)变量：$\{1, 2, 3, 4, 5\}$

   　区组：$\{1, 2, 3, 4\}$，$\{1, 3, 4, 5\}$，$\{1, 2, 4, 5\}$，$\{1, 2, 3, 5\}$，$\{2, 3, 4, 5\}$

   (d)变量：$\{1, 2, 3, 4, 5, 6, 7\}$

   　区组：$\{1, 2, 3\}$，$\{1, 2, 4\}$，$\{1, 3, 5\}$，$\{1, 4, 6\}$，$\{1, 5, 7\}$，$\{1, 6, 7\}$，$\{3, 5, 6\}$
   $\{2, 3, 7\}$，$\{2, 4, 5\}$，$\{2, 5, 6\}$，$\{2, 6, 7\}$，$\{3, 4, 6\}$，$\{3, 4, 7\}$，$\{4, 5, 7\}$

   (e)变量：$\{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p\}$，

   　区组：$\{a, b, c, d, e, f\}$，$\{a, b, g, h, i, j\}$，$\{a, c, g, k, l, m\}$，$\{a, d, h, k, n, o\}$，
   $\{a, e, i, l, n, p\}$，$\{a, f, j, m, o, p\}$，$\{b, c, g, n, o, p\}$，$\{b, d, h, l, m, p\}$，
   $\{b, e, i, k, m, o\}$，$\{b, f, j, k, l, n\}$，$\{c, d, i, j, m, n\}$，$\{c, e, h, j, k, p\}$，
   $\{c, f, h, i, l, o\}$，$\{d, e, g, j, l, o\}$，$\{d, f, g, i, k, p\}$，$\{e, f, g, h, m, n\}$　543

2. (a)一个BIBD有参数$v=6$，$k=3$，$\lambda=10$，确定b和r．

   (b)一个BIBD有参数$v=13$，$b=78$，$r=24$，确定k和λ．

   (c)一个BIBD有参数$b=85$，$k=21$，$\lambda=5$，确定v和r．

3. 证明不存在有下面参数的(b, v, r, k, λ)设计：

   (a)$b=6$，$v=9$，$r=2$，$k=3$，$\lambda=1$　　(b)$b=22$，$v=22$，$r=7$，$k=7$，$\lambda=2$

4. 证明不存在有参数$b=4$，$v=9$，$r=4$，$k=9$，$\lambda=4$的(b, v, r, k, λ)设计．
5. 可能有$(12, 6, 8, 7, 1)$设计吗？
6. 可能有$(13, 3, 26, 6, 1)$设计吗？
7. 可能有$(6, 8, 6, 8, 6)$设计吗？
8. 在Wadley实验(例9.13)中，对于有5头牛的情况，确定λ．
9. 对于练习1中的每一个区组设计，确定它的关联矩阵．
10. 对于下面每一个区组设计及该设计的关联矩阵$\boldsymbol{A}$，计算$\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$．

    (a)例9.11的设计

    (b)例9.12的设计

    (c)$(15, 15, 7, 7, 3)$设计

11. 确定$v=9$的施泰纳三元系统中的b和r．

12. 下面 9 个区组构成有 9 个变元的施泰纳三元系统设计的一部分：

$$\{a,b,c\},\{d,e,f\},\{g,h,i\},\{a,d,g\},\{c,e,h\},\{b,f,i\},\{a,e,i\},\{c,f,g\},\{b,d,h\}.$$

(a)缺少多少个区组?

(b)额外增加区组使得产生一个施泰纳三元系统.

13. 说明为什么由例 9.9 中的正交阵列构造的门限方案不是匿名的.
14. 存在一个可分解(35, 15, 7, 3, 1)设计. 确定平行类的数量及每一个平行类中的区组的数量.
15. 确定一个不同于表 9.25 中设计的可分解(12, 9, 4, 3, 1)设计.
16. X 和 X' 上的两个设计是同构的(isomorphic)，如果存在从 X 到 X' 上的一一映射函数 f，使得只要 B 是 X 的一个区组则 $f(B)=\{f(x) \mid x\in B\}$ 是 X' 的一个区组.

(a)证明任意两个可分解(12, 9, 4, 3, 1)设计是同构的.

(b)证明 7 个变元集合的任意两个施泰纳三元系统是同构的.

17. 在基于表 9.25 的设计的一个匿名门限方案中，领导者给人 u 部分信息 8 而给人 v 部分信息 6. 这一计划的密钥是什么?
18. 如果一个施泰纳三元系统有 67 个变元，那么它有多少个区组?
19. 对于有 13 个变元的施泰纳三元系统的关联矩阵 $\boldsymbol{A}$，计算 $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$. [544]
20. 给定一个设计，补设计(complementary design)的关联矩阵可以通过交换原来设计的关联矩阵中的 0 和 1 而得到. 一般地，假设原来设计是(b, v, r, k, λ)设计，补设计是$(b', v', r', k', \lambda')$设计.

(a)确定关于 b'，v'，r'和 k'的公式.

(b)证明 $\lambda'=b+\lambda+2r$.

(c)确定一个(16, 16, 6, 6, 2)设计.

(d)确定一个(12, 9, 8, 6, 5)设计.

21. 假设一个有 13 个变元的施泰纳三元系统的补设计(练习 20)是(b, v, r, k, λ)设计. 确定 b，v，r，k，λ.
22. 构造一个有 21 个变元的施泰纳三元系统.
23. 通过证明集合 S 实际上是一个施泰纳三元系统来完成定理 9.15 的证明.
24. (7, 3, 1)设计的 4 个区组是

$$\{1,2,3\},\quad \{1,5,6\},\quad \{2,5,7\},\quad 和\quad \{1,4,7\}.$$

确定余下的区组.

25. 通过构造法证明存在$(v, v-1, v-2)$设计.
26. 证明下面的每一个设计存在.

(a)A(31, 15, 7)设计　　(b)A(63, 31, 15)设计

(c)A(21, 5, 1)设计　　(d)A(31, 6, 1)设计

27. 对于(31, 15, 7)设计的关联矩阵 $\boldsymbol{A}$，计算 $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$.
28. 证明在(v, k, λ)设计中，任意两个区组都正好有 λ 个公共变元.
29. (a)如果 $\boldsymbol{A}$ 是(b, v, r, k, λ)设计的关联矩阵，证明 $\boldsymbol{A}^{\mathrm{T}}$ 不一定是(v, k, λ)设计的关联矩阵.

(b)证明：如果 $\boldsymbol{A}$ 是(v, k, λ)设计的关联矩阵，那么 $\boldsymbol{A}^{\mathrm{T}}$ 是(v, k, λ)设计的关联矩阵.

30. 证明不存在(43, 43, 7, 7, 1)设计.
31. 考虑整数$\{0, 1, 2, \cdots, v-1\}$的集合 Z_v，以及模 v 的加法. Z_v 中 k 个整数的子集 D 称为(v, k, λ)*差集*((v, k, λ)-difference set)，或简称*差集*，当我们列出 D 中不同元素 x，y 的差(同时使用 $x-y$ 和 $y-x$)模 v 时，Z_v 中的每一个非 0 整数正好都出现同样的 λ 次.

(a)证明 $D=\{0, 1, 3\}$是 Z_7 的差集.

(b)证明 $D=\{0, 1, 4\}$不是 Z_7 的差集.

(c)证明 $D=\{0, 1, 6, 8, 18\}$是 Z_{21}的差集.

(d)确定一个(11, 5, 2)差集.

(e)如果 D 是(v, k, λ)差集，那么它有多少个元素?

(f)如果 D 是一个(v, k, λ)差集，确定以 v 和 k 为参数的函数 λ 的表达式.

32. 假设 D 是一个(v, k, λ)差集合. 如果 $x \in Z_v$，设

$$D+x=\{y+x: y \in D\},$$

其中的加法是模 v 加法.

545

(a)证明下面的定理：

定理 如果 D 是一个(v, k, λ)差集，那么$\{D+x: x \in Z_v\}$是一个(v, k, λ)设计.

(b)对应于 Z_7 中的差集 $D=\{0, 1, 3\}$，通过构造一个(7, 3, 1)设计来说明这一定理.

(c)对应于 Z_{21} 中的差集 $D=\{0, 1, 6, 8, 18\}$，通过构造一个(21, 5, 1)设计来说明这一定理.

(d)对应于练习 31(d)中所确定的差集，通过构造一个(11, 5, 2)设计来说明这一定理.

33. 证明如果 $m \geqslant 1$ 是一个素数的幂，那么存在$(2m^2+2m+2, m^2+m+1, 2m+2, m+1, 2)$设计.

34. 使用 Bruck-Ryser-Chowla 定理证明(20, 5, 1)设计可能存在.

35. 下列的各(v, k, λ)设计中哪些可能存在？

(a)(16, 9, 1)　　(b)(34, 12, 4)　　(c)(43, 7, 1)

36. 证明(46, 46, 10, 10, 2)设计不存在.

37. 通过构造法证明存在(14, 8, 7, 4, 3)设计. [提示：使用定理 9.17 和另一个定理.]

38. 通过构造法证明存在(30, 16, 15, 8, 7)设计. [提示：使用定理 9.17 和另一个定理.]

39. 证明存在(30, 15, 14, 7, 6)设计.

40. 假设存在(v, k, λ)设计.

(a)证明存在$(2v, v, 2k, k, 2\lambda)$设计.

(b)证明对于任意正整数 p，存在$(pv, v, pk, k, p\lambda)$设计.

41. 证明存在(62, 31, 30, 15, 14)设计.

42. 我们希望测试由 6 个人组成的组群中 HIV 的携带者，这 6 个人的名字记作 A, B, C, D, E, F. 设 $P \subseteq U=\{A, B, C, D, E, F\}$. 我们使用 4 个组群 $X_1=\{A, B, C\}$, $X_2=\{A, D, E\}$, $X_3=\{B, D, F\}$, $X_4=\{C, E, F\}$.

(a)对于 U 的所有子集 P，计算向量$(f_{X_1}(P), f_{X_2}(P), f_{X_3}(P), f_{X_4}(P))$.

(b)如果向量(1, 1, 1, 1)是某个 P 的 $f_G(P)$，而且我们知道 $|P| \leqslant 2$，我们能够确定 P 吗？

(c)证明这一组群的集合给出有门限值 1 的成功 NAGTA.

43. 我们希望确定 8 家有线电视提供商的新网络的效益，这 8 家提供商的名字记为 A, B, …, H. 设 P 包含于{A, B, C, D, E, F, G, H}. 考虑由下面 6 个组群组成的集合 G, $X_1=\{A, B, C, D\}$, $X_2=\{E, F, G, H\}$, $X_3=\{A, C, E, G\}$, $X_4=\{B, D, F, H\}$, $X_5=\{A, B, D, G\}$, $X_6=\{C, E, F, H\}$.

(a)如果向量(1, 0, 1, 0, 1)是某个 P 的 $f_G(P)$，你能确定 P 吗？

546

(b)证明集合 G 无法给出有门限值 1 的成功 NAGTA.

44. (a)使用表 9.25 的(12, 9, 4, 3, 1)设计构造一个有门限值 3 的成功 NAGTA.

(b)平行类 C_i 中的第 j 个区组记作 $B_{i,j}$. 如果

$$f_G(P)=(1,1,0,1,1,0,1,0,1,1,1,1)$$

且我们知道 $|P| \leqslant 3$，确定 P.

45. 如果 $t \geqslant 2$，一个 t-(b, v, r, k, λ)设计由 $v \geqslant 2$ 个变元的集合 V 以及称为区组的 V 的 $b>0$ 个子集的集合组成，使得(9.10)式和(9.11)式成立，且有

$$V \text{ 的每个 } t \text{ 子集正好是 } \lambda \text{ 个区组的子集}, \quad \lambda>0, \tag{9.29}$$

而且 $k<v$. 显然，2-(b, v, r, k, λ)设计是(b, v, r, k, λ)设计.

(a)假设 $x_{i_1}, x_{i_2}, \cdots, x_{i_t}$ 是 t-(b, v, r, k, λ)设计的 t 个不同的变元. 对于 $1 \leqslant j \leqslant t$，设 λ_j 是包含 $x_{i_1}, x_{i_2}, \cdots, x_{i_j}$ 的区组的数量. 设 $\lambda_0=b$. 证明对于 $0 \leqslant j \leqslant t$，有

$$\lambda_j = \frac{\lambda\binom{v-j}{t-j}}{\binom{k-j}{t-j}}, \tag{9.30}$$

且得出结论：λ_j 与 x_{i_1}，x_{i_2}，…，x_{i_j} 的选取无关. 因此得出结论：对于所有的 $1\leqslant j\leqslant t$，t-(b, v, r, k, λ)设计也是 j-(b, v, r, k, λ)设计.

(b)证明对于 $t\geqslant 2$，(9.10)式和(9.29)式蕴涵(9.11)式.

(c)注意，如果 t-(b, v, r, k, λ)设计存在，那么对于所有的 j，由(9.30)式所定义的数 λ_j 是整数，且有$0\leqslant j\leqslant t$.

(d)9.5.2 节的结果表明不存在(43，7，1)设计. 使用这一结果证明即使 λ_j 是整数，这仍然不是 t-(b, v, r, k, λ)设计存在的充分条件.

46. 假设方阵 $\boldsymbol{A}$ 是一个 BIBD 的关联矩阵. 证明 $\boldsymbol{A}^{-1}$存在.

47. 如果 $\boldsymbol{A}$ 是(b, v, r, k, λ)设计的关联矩阵，证明 $\boldsymbol{AJ}=r\boldsymbol{J}$，其中 $\boldsymbol{J}$ 是所有项均为 1 的矩阵.

48. 假设 $\boldsymbol{A}$ 是 0 和 1 组成的 $v\times v$ 矩阵，$v\geqslant 2$，且存在$k>0$ 和 $\lambda>0$，且 $k>\lambda$，使得：

(1)$\boldsymbol{A}$ 的任意行正好包含 k 个 1.

(2)$\boldsymbol{A}$ 的任意一对行正好在λ 个列上都为 1.

这一练习要求读者证明：

(3)$\boldsymbol{A}$ 的任意列正好包含 k 个 1.

547 (4)$\boldsymbol{A}$ 的任意一对列正好在λ 个行上都为 1.

[特别地，这表明对于(v, k, λ)设计的关联矩阵(3)和(4)成立，且定理 9.19 成立.]

(a)证明 $\boldsymbol{AJ}=k\boldsymbol{J}$，其中 $\boldsymbol{J}$ 是所有项均为 1 的方阵.

(b)证明 $\boldsymbol{AA}^{\mathrm{T}}=(k-\lambda)\boldsymbol{I}+\lambda\boldsymbol{J}$.

(c)证明 A^{-1}存在.

(d)证明 $\boldsymbol{A}^{-1}\boldsymbol{J}=k^{-1}\boldsymbol{J}$.

(e)证明 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}=(k-\lambda)\boldsymbol{I}+\lambda k^{-1}\boldsymbol{JA}$.

(f)证明如果 $\boldsymbol{JA}=k\boldsymbol{J}$，那么(3)和(4)成立.

(g)证明 $\boldsymbol{JA}=k^{-1}(k-\lambda+\lambda v)\boldsymbol{J}$.

(h)证明 $k-\lambda+\lambda v=k^2$.

(i)证明 $\boldsymbol{JA}=k\boldsymbol{J}$，因此(3)和(4)成立.

49. 在一项实验中，存在两种处方或变元，即控制和非控制. 存在 3 个控制和 120 个区组. 每一个控制用于 48 个区组中. 而每一对控制用于相同的区组 24 次. 所有 3 个控制一起用于相同区组中 16 次. 不使用任何控制的区组有多少个?

50. (Stison[2003])假设有一个匿名(2，p)门限方案，它允许任意两个人能确定密钥，但是任何一个人找到密钥的概率都不能大于 $1/|K|$.

(a)证明 $|P|\geqslant(p-1)|K|+1$. [提示：对于 $k\in K$，当密钥是 k 时，设 C_k 是可以分配的 P 的所有可能的 p 元素子集，一个人一个子集合. 证明这些子集正好在一个元素上重叠.]

(b)证明如果 $|P|=(p-1)|K|+1=v$，那么一定存在一个 $k=p$ 的可分解(b, v, r, k, λ)设计. [提示：以(a)中定义的子集为区组.]

51. 证明不存在 $S(5, 7, 18)$施泰纳系统.

52. 在一个 $S(5, 6, 48)$施泰纳系统中有多少个区组?

53. 假设存在集合 V 上的 $S(t, k, v)$施泰纳系统.

(a)证明 $S(t-1, k-1, v-1)$施泰纳系统也存在. [提示：固定 V 中的一个元素，并只考虑包含这个元素的那些 k 元素子集.]

(b)对于每一个 $j<t$，证明 $S(t-j, k-j, v-j)$施泰纳系统存在.

54. (Anderson[1990])使用练习53及在集合V上存在$S(5,8,24)$施泰纳系统的事实．证明：

(a)$S(5,8,24)$施泰纳系统的区组数是759.

(b)V中的每一个元素存在于253个区组中.

(c)V中的每一对元素存在于77个区组中.

(d)V的每三个元素存在于21个区组中.

(e)V中的每四个元素存在于5个区组中.

(f)V中的每五个元素存在于1个区组中.

548

9.5 有限射影平面

9.5.1 基本性质

有趣的是实验设计有几何学应用，而反过来，几何学又在实验设计的分析中起着重要的作用．我们考虑例9.10的设计．这是一个施泰纳三元系统而且是一个对称的BIBD．通过把变元设成点，并使用经过点的"线"(不一定是直线)表示区组，我们可以用几何方式来表示这个设计．图9.1给出这一几何表示．除一条线外其余线都是直线．这一表示称为**射影平面**(projective plane)或**费诺平面**(Fano plane)[⊖]．这一表示有下面的性质：

图9.1 费诺平面

(P_1)两个不同的点位于一条且只位于一条公共线上.

(P_2)两条不同的线经过一个且只经过一个公共点.

一般地，**射影平面**由下面几部分组成：称为**点**的对象集合，称为**线**的另一个对象集合，以及使得条件(P_1)和(P_2)成立的**点位于线上**，或等价地，**线经过点**的概念．射影平面是**有限的**(finite)，如果点的集合是有限的．射影平面不仅在组合设计中很重要，而且在艺术领域也很重要，在艺术领域，对投影进行研究时就出现射影平面．它们在几何学中也很重要，因为它们定义了一种欧几里得平行公设不成立的几何：根据(P_2)，不存在经过给定点且与给定线没有公共点(因此与其"平行")的线．射影几何的发展可追溯到4世纪Pappas of Alexandria的工作．在19世纪40年代，它导致了由著名数学家Boole、Cayley和Sylvester创建的不变量代数理论的产生．它又导致了张量演算的产生，最终促进了物理学中的基础思想的产生，特别是对重力理论中爱因斯坦的工作产生了影响.

组合设计理论中重要的基本存在问题也发生于射影平面：对于什么样的值n存在有n个点的射影平面？如果$n=2$，我们可以取两个点a和b，和一条经过这两个点的线L．射影平面的公设(P_1)和(P_2)显然满足．对于任意的n，如果存在n个点和一条经过所有n个点的线，显然也满足这些公设．最后，当有三个点a，b，c，和三条线L_1，L_2，L_3，且a和b位于L_1上，b和c位于L_2上，a和c位于L_3上时，这些公设显然也满足．为了剔除这些平凡例子，通常我们要加入一条额外的公设：

(P_3)存在4个不同的点，其中任意3个点不在同一条线上.

满足(P_3)的有限射影平面称为**非退化的**(nondegenerate)，而且我们假设(不再每一次都明确做此假设)所有**限射影平面都是非退化的**．任意关于这些平面的定理都将使用这三条公设(P_1)、

549

⊖ 以19世纪数学家Gino Fano的名字命名.

(P_2)和(P_3)进行证明.

最小可能的射影平面至少有四个点. 存在正好有四个点的这样的平面吗? 假设a, b, c, d是四个点, 且任意三个点都不在一条线上. 根据(P_1), 一定存在经过a和b的线L_1, 和经过c和d的线L_2, 因为没有三个点在一条线上, 所以c和d不在L_1上, 且a和b不在L_2上. 于是, 如果a, b, c和d是这个射影平面的所有点, 那么L_1和L_2没有公共交点, 这与(P_2)不符. 因此, 不存在四个点的射影平面. 下面我们将看到也不存在5个点或6个点的射影平面. 然而, 图9.1的费诺平面是一个7个点的射影平面, 因为(P_3)很容易证明.

读者将注意到公设(P_1)和(P_2)有某种形式的*对偶性*(duality): 通过交换词汇"点"和"线", 并交换表达方式"位于线上的点"和"经过一个点的线", 我们可以从(P_1)得到(P_2). 通过同样的交换我们可以从(P_2)得到(P_1). 如果(P)是任意关于有限射影平面的*陈述*(statement), 那么(P)的*对偶*(dual)是通过做这些交换从(P)而得的陈述. 可以证明公设(P_3)的对偶为真, 我们把这一结果阐述为一个定理.

定理 9.23 *在有限射影平面内, 下面的陈述成立:*

(P_4)存在四条不同的线, 其中没有三条线经过相同的点.

证明 根据(P_3), 存在点a, b, c, d, 其中没有三个点在一条线上. 根据(P_1), 存在经过a和b的L_1, 经过b和c的线L_2, 经过c和d的线L_3, 经过d和a的线L_4. 现在, 这四条线是不同的, 因为c和d不在L_1上, a和d不在L_2上, 等等. 另外, 没有三条线经过一个公共点. 我们将通过矛盾法证明这一定理. 不失一般性, 假设L_1, L_2和L_3有公共点x. 那么x不是b, 因为b不在L_3上, 此时, L_1和L_2有两个不同的公共点b和x. 因为$L_1 \neq L_2$, 公设(P_2)不成立, 这是一个矛盾. ∎

550

现在, 条件(P_1)和(P_2)是对偶的, 条件(P_3)和(P_4)是对偶的. 任意关于有限射影平面的定理(可证明的陈述)必须通过公设(P_1), (P_2)和(P_3)进行证明. 任意这样的定理有*对偶定理*(dual theorem), 这一对偶定理是通过交换单词"点"和"线"以及交换表述"位于线上的点"和"经过点的线"而得到的. 对偶定理的证明可以通过原定理的证明得到, 其方法是使用(P_1), (P_2)和(P_3)的对偶陈述取代(P_1), (P_2)和(P_3), 我们知道这样的对偶陈述是成立的. 因此, 我们有下面的结果.

定理 9.24(对偶原理) *对于每一个是定理的关于有限射影平面的陈述, 其对偶陈述也是定理.*

关于射影平面的下一个基本定理如下所述.

定理 9.25 *在有限射影平面内, 每一个点位于相同数量的线上, 每一条线经过相同数量的点.*

为了说明这一定理, 我们注意到, 图9.1的射影平面的每一条线上有三个点, 而每一个点经过三条线.

定理 9.25 的证明 首先, 我们证明每一条线经过相同数量的点. 证明的基本思想是建立两条不同线L和L'上的点之间的一一对应, 这一对应表明两条线上有相同数量的点.

我们先证明存在既不在L上也不在L'上的点x. 根据公设(P_3), 存在四个点a, b, c, d, 其中没有三个点在一条线上. 如果其中任何一个点不在L或L'上, 那么我们就可以取那个点为x. 如果这些点都在L或L'上, 那么我们一定可以在L上得到两个点(比如说是a和b), 且在L'上也得到两个点(比如说是c和d). 根据(P_1), 存在经过b和c的线K和经过a和d的线K'. 根据(P_2), 线K和K'有一个公共点x(参见图9.2). 如果x位于L上, 那么x和b位于两条不同的线上, 违反(P_1). 如果x位于L'上, 那么x和c位于两条不同的线上. 也违反(P_1). 因此, x是符合我们要求的点.

551

现在, 给定线L上的点p, 这条经过p和x的线[根据(P_1)一定存在这样的线]必定与L'正

好相交于一个点 p'[根据(P_2)]. 我们说 p' 是 p 经过 x 在 L' 上的投影点(参见图 9.3). 如果 q 是 L 上的其他任意点，设 q' 是 q 经过 x 在 L' 上的投影点. 现在，如果 $q\neq p$，那么 q' 一定不同于 p'. 因为否则 q' 和 x 在两条不同的线上，违反(P_1). 我们得出结论：投影定义 L 和 L' 上的点之间的一一对应. 我们知道这是一对一的. 为了弄明白它是一个对应，注意，L' 上的每个点 r 都可以通过上述过程从 L 上的某个点得到. 为了弄明这一点，只需从 r 经过 x 投影回到 L 上. 因此，L 和 L' 有相同数量的点. 这就证明了每一条线经过相同数量的点.

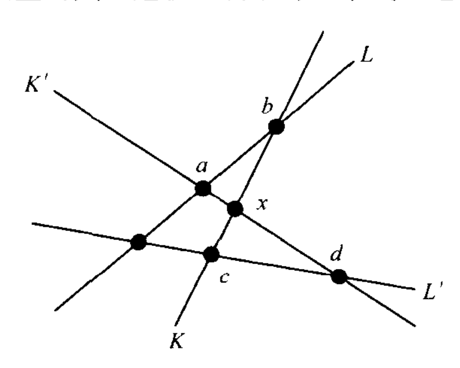


图 9.2 点 x 既不在 L 也不在 L' 上

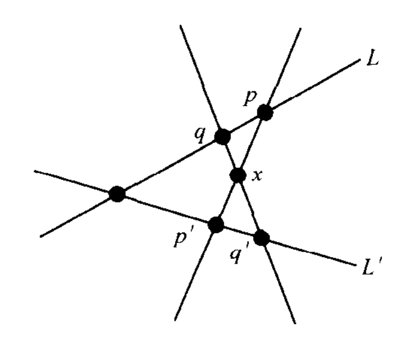


图 9.3 点 p' 是 p 经过 x 在 L' 上的投影点

通过利用对偶原理(定理 9.24)，我们得到每一个点位于相同数量的线上. ■

定理 9.26 *在有限射影平面内，经过每一个点的线的数量等于每一条线上点的数量.*

证明 选出任意一条线 L. 根据(P_3)，存在一个不在 L 上的点 x. 根据(P_1)，对于 L 上的任意点 y，存在且仅存在一条经过 x 和 y 的线 $L(y)$. 另外，经过 x 的任意线 L' 不是 L，因此，根据(P_2)，这条线一定经过 L 上的一个点 y，所以 L' 是 $L(y)$. 因此，$L(y)$ 定义 L 上的点与经过 x 的线之间的一一对应. 因此，经过 x 的线的数量等于 L 上点的数量. 根据定理 9.25，这一定理成立. ■

因此根据定理 9.26，在每条线上有 $m+1$ 个点的射影平面有 $m+1$ 条线经过每一个点.

推论 9.26.1 *每一条线上有 $m+1$ 个点且经过每一个点有 $m+1$ 条线的射影平面有 m^2+m+1 个点和 m^2+m+1 条线.*

552

证明 设 x 是任意点. 存在 $m+1$ 条经过 x 的线. 每一条这样的线有 m 个不同于 x 的点. 每一个点位于一条且仅一条经过 x 的线上. 因此，我们可以通过计数经过 x 的线上又不同于 x 的点的数量再加上 x 而得到点的数量；我们得到

$$(m+1)(m)+1=m^2+m+1$$

个点. 推论的其余部分可以根据对偶原理得到. ■

在图 9.1 的例子中，$m+1=3$，$m=2$，$m^2+m+1=7$. 我们现在知道，只对形如 m^2+m+1 的 n 存在 n 个点的射影平面. 特别地，对于 $n=4$，5，6，不存在这样的射影平面. 公设(P_3)剔除 $n=3$ 的情况(既使 $3=1^2+1+1$). 因此，$n=7$ 对应于第一个可能的射影平面. 下一个可能的射影平面通过取 $m=3$ 得到，此时 $n=3^2+3+1=13$. 我们将看到，每当 m 是素数的幂时总是存在射影平面. 因此，存在 13 个点的射影平面($m=3$). (表 9.30 给出一个这样的平面.)数 m 将被称为射影平面的秩(order). 注意这个秩与点的数量不同.

表 9.30 秩为 3(有 13 个点和 13 条线)的射影平面

点：	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13
线：	{1, 2, 8, 12}, {3, 9, 10, 12}, {4, 5, 11, 12}, {1, 4, 6, 9}, {3, 6, 8, 11}, {2, 5, 6, 10}, {1, 10, 11, 13}, {5, 8, 9, 13}, {2, 3, 4, 13}, {1, 3, 5, 7}, {4, 7, 8, 10}, {2, 7, 9, 11}, {6, 7, 12, 13}.

9.5.2 射影平面、拉丁方和(v, k, λ)设计

本节研究射影平面与某种组合设计之间的关系.

定理 9.27(Bose[1938]) *假设 $m\geqslant 2$. 存在秩为 m 的有限射影平面，当且仅当存在完全正交 $m\times m$ 拉丁方族.* ⊖

这一定理的证明是构造性的；即给出如何在有限射影平面与正交拉丁方集合之间转换. 我们将在练习 22 和 23 概述此定理的证明.

推论 9.27.1 *对于素数 p 和正整数 k，如果有 $m=p^k$，那么存在秩为 m 的射影平面.*

553 **证明** 根据定理 9.2，这一结果成立. ■

根据定理 9.27 和推论 9.27.1，可得出第一个不存在射影平面的可能秩 m(即 m^2+m+1 个点的射影平面)的值是 6. 事实上，对于 $m=6$，我们已在 9.2 节中看到，不存在 5 个 6×6 正交拉丁方集合(实际上不存在这样的拉丁方对)，因此，不存在秩为 6 的有限射影平面(即有 $6^2+6+1=43$ 个点的射影平面). 存在秩为 7，8 和 9 的射影平面，因为 $7=7^1$，$8=2^3$，$9=3^2$. 然而，$m=10$ 不是素数的幂. Lan，Thiel 和 Swiercz[1989]使用计算机证明了不存在秩为 10 的有限射影平面(即 $10^2+10+1=111$ 个点的射影平面.)我们知道存在秩为 11 的有限射影平面，但是对于 12 至今仍是一个未解问题.

下面的定理处理推论 9.27.1 没有涵盖的某些情况. 我们略去其证明.

定理 9.28(Bruck 和 Ryser[1949]) *设 $m\equiv 1$ 或 $2 \pmod 4$. 假设能整除 m 的最大平方是 d 且 $m=m'd$. 如果 m' 能被素数 p 整除，且 $p\equiv 3 \pmod 4$，那么不存在秩为 m 的射影平面.*

为了说明这一定理，假设 $m=6$. 注意 $6\equiv 2 \pmod 4$. 于是，$d=1$ 且 $m'=m$. 因为 m' 能被 3 整除，正如我们前面所观察的那样，不存在秩为 6 的射影平面. 接下来，假设 $m=54$. 注意 $54\equiv 2 \pmod 4$，$d=9$，$m'=6$. 因为 m' 能被 3 整除，所以不存在秩为 54 的射影平面. 根据定理 9.27 可得不存在秩为 54 的完全正交拉丁方族.

通过取点作为变元，而取线作为区组，一个射影平面给出一个(b, v, r, k, λ)设计. 于是 $b=v=m^2+m+1$，$k=r=m+1$，$\lambda=1$，因为每一对点都位于一条且仅在一条线上，所以我们得到一个对称平衡不完全区组设计或(v, k, λ)设计. 相反，对于每一个 $m\geqslant 2$，(m^2+m+1, $m+1$, 1)设计给出一个射影平面，其中取变元为点，区组为线. 为了弄明白为什么公理(P_3)成立，设 a 和 b 是任意两个点. 存在唯一包含 a 和 b 的区组 B_1. 因为这一设计是不完全的，所以存在不在区组 B_1 内的点 x. 现在，存在唯一包含 a 和 x 的区组 B_2，且存在唯一包含 b 和 x 的区组 B_3. 每个区组有 $m+1$ 个点. 因此，除了 a，b 和 x 之外，B_1，B_2 和 B_3 每一个都至多有 $m-1$ 个其他点. 简而言之，B_1，B_2 或 B_3 中的点的数量至多是 $3+3(m-1)=3m$. 因为 $m\geqslant 2$，我们有 $m^2+m+1>3m$. 因此，存在不在 B_1，B_2 和 B_3 内的点 y. 点 a，点 b，点 x，点 y 中没有三个点在一个区组内. 因此我们有下面的结果.

定理 9.29 *如果 $m\geqslant 2$，存在秩为 m 的有限射影平面当且仅当存在(m^2+m+1, $m+1$, 1)设计.*

推论 9.29.1 *只要 m 是一个素数的幂，那么存在(m^2+m+1, $m+1$, 1)设计.*

554 **推论 9.29.2** *假设 $m\geqslant 2$. 那么下面各陈述等价.*

(a)存在秩为 m 的有限射影平面.

(b)存在秩为 m 的完全正交拉丁方族.

(c)存在(m^2+m+1, $m+1$, 1)设计.

⊖ 回想一下，根据我们的习惯，单一 2×2 拉丁方构成一个正交族.

本节练习

1. 在下列每一种情况中，P 是点的集合，Q 是线的集合. 点 x 位于线 L 上，如果 $x \in L$，线 L 经过点 x，如果 $x \in L$. 确定公理(P_1)，(P_2)和(P_3)中哪一个成立.
   (a)P=3 维空间中所有的点，Q=3 维空间中的所有线.
   (b)$P=\{1, 2, 3\}$，$Q=\{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$.
   (c)P=任意集合，Q=P 的元素的所有序对.
   (d)P=3 维空间的所有线，Q=3 维空间的所有平面.
   (e)P=经过 3 维空间原点的所有线，Q=经过 3 维空间原点的所有平面.
2. 给出下列每一个(不一定为真)关于有限射影平面的陈述的对偶陈述.
   (a)存在 9 条不同的线，其中没有 3 条线经过同一点.
   (b)存在位于每一条线上的点.
   (c)存在四条不同的线使得每一个点位于其中的一条线上.
   (d)存在四个不同点，其中没有三个点位于同一条线上，使得每一条线经过其中的一个点.
3. 如果一个射影平面中每一条线上有四个点. 这个射影平面总共有多少个点?
4. 如果一个射影平面中每一点有 5 条线通过，这个射影平面总共有多少个点?
5. 如果一个射影平面有 31 个点，那么每一条线上有多少个点?
6. 如果一个射影平面有 57 个点，那么每一个点有多少条线通过?
7. 如果一个射影平面有 73 条线，那么每一条线上有多少个点?
8. 存在下面的射影平面吗?
   (a)25 个点　　(b)73 个点　　(c)43 条线　　(d)91 条线
9. 假设射影平面有 n 个点，且(v, k, λ)设计是由点为变元、线为区组的平面定义的. 对于下列每一个 n 的值，计算 v，k 和 λ 的值.
   (a)31　　(b)91　　(c)133
10. 是否存在秩为下列各值的有限射影平面? 验证你的答案.
   (a)11　　(b)49　　(c)81

555

11. 证明不存在秩为 14 的有限射影平面.
12. 证明不可能存在秩为 245 的有限射影平面.
13. 证明不可能存在秩为 150 的有限射影平面.
14. 能否存在秩为下列各值的有限射影平面? 验证你的答案.
   (a)60　　(b)81　　(c)93
15. 证明不存在秩为 378 的完全正交拉丁方族.
16. 回想定理 9.25 的证明中**投影**(projection)的定义. 在费诺平面中(图 9.1):
   (a)确点线{3, 4, 6}上的点 6 经过点 5 在线{1, 3, 7}上的投影.
   (b)确定线{1, 2, 4}上的点 4 经过点 3 在线{2, 6, 7}上的投影.
   (c)确定线{1, 2, 4}上的点 2 经过点 3 在线{4, 5, 7}上的投影.
17. 回想定理 9.25 的证明中投影的定义. 在表 9.30 的射影平面内:
   (a)确定线{2, 3, 4, 13}上的点 2 经过点 8 在线{1, 4, 6, 9}上的投影.
   (b)确定线{2, 7, 9, 11}上的点 11 经过点 1 在线{2, 5, 6, 10}上的投影.
18. (Bogart[1983])取(n^2+n, n^2, $n+1$, n, 1)设计的点为点，而这个设计区组为线. 公理(P_1)、(P_2)和(P_3)中哪一个满足?
19. 证明在练习 18 中，下面的“平行公设”满足：给定点 x 和不经过 x 的线 L，存在且仅存在一条经过 x 且与 L 没有公共点的线 L'.
20. **仿射平面**(affine plane)与射影平面相仿，它满足公理(P_1)、(P_3)和练习 19 的“平行公设”. 证明下列关

于仿射平面的陈述.

(a)每一条线包含相同数量的点.

(b)每一个点在相同数量的线上.

(c)如果每一条线都包含相同数量的 n 个点，那么每一个点正好在 $n+1$ 条线上，正好存在 n^2 个点和 n^2+n 条线.

21. 在一个仿射平面内(练习 20)，每一条线包含相同数量的 n 个点. 我们称 n 为这个仿射平面的秩. 证明如果存在秩为 n 的射影平面，那么存在秩为 n 的仿射平面[提示：消除一条线和其上的所有点].

556

22. 以下两个练习给出定理 9.27 的证明概况. 设 P 是秩为 m 的有限射影平面.

(a)从 P 中任意选出一条线 L，并称 L 为无穷远线(line at infinity). 设 L 有点：

$$u,v,w_1,w_2,\cdots,w_{m-1}.$$

存在 $m+1$ 条线经过 L 的每一个点，因此是 m 条线外加 L. 设这些线为如下所列的线：

通过 u 的线：U_1，U_2，…，U_m，

通过 v 的线：V_1，V_2，…，V_m，

通过 w_j 的线：W_{j1}，W_{j2}，…，W_{jm}.

不在 L 上的每一个点 x 与 L 的每一个点通过唯一一条线相连. 假设 U_h 是包含 x 和 u 的线，V_i 是包含 x 和 v 的线，W_{jk_j} 是包含 x 和 w_j 的线. 因此，我们可以把点 x 与 $(m+1)$ 元组 $(h, i, k_1, k_2, \cdots, k_{m-1})$ 联系起来. 证明不在 L 上的点 x 与序对 (h, i) 间的对应是一一对应.

(b)利用表 9.30 的射影平面说明这一构造法. 假设{1, 2, 8, 12}是无穷远线，写出所有线和关系 $x\to(h, i)$ 和 $x\to(h, i, k_1, k_2)$.

(c)如果对应于序对 (h, i) 的点 x 给出 $(m+1)$ 元组 $(h, i, k_1, k_2, \cdots, k_{m-1})$，设 $a_{hi}^{(j)}=k_j$，且设 $A(j)=(a_{hi}^{(j)})$，$j=1, 2, \cdots, m-1$. 对于表 9.30 的射影平面，确定 $A^{(1)}$ 和 $A^{(2)}$.

(d)证明 $A^{(j)}$ 是拉丁方.

(e)证明：如果 $q\neq p$，则 $A^{(p)}$ 和 $A^{(q)}$ 是正交的.

23. 假设 $A^{(1)}$，$A^{(2)}$，…，$A^{(m-1)}$ 是秩为 m 的两两正交的拉丁方族.

(a)考虑 m^2 个"有限"点 (h, i)，$h=1, 2, ..., m$，$i=1, 2, ..., m$. 给定点 (h, i)，将它与 $(m+1)$ 元组

$$(h,i,k_1,k_2,\cdots,k_{m-1})$$

相联系，其中 k_j 是 $a_{hi}^{(j)}$. 给定表 9.14 所示的秩为 3 的两个正交拉丁方 $A^{(1)}$ 和 $A^{(2)}$，确定这些 $(m+1)$ 元组. (这将是我们的示例.)

(b)通过设 W_{jk} 是所有有限点 (h, i) 的集合，其中对应于 (h, i) 的 $(m+1)$ 元组中的第 $(j+2)$ 项是 k，构成 $m^2+m=m(m+1)$ 条线 W_{jk}，$j=-1, 0, 1, 2, \cdots, m-1$，$k=1, 2, \cdots, m$(这些线将通过后面的一个点被延长). 确定示例中的这些线 W_{jk}.

(c)注意，对于固定的 j，我们定义的

$$W_{j1},W_{j2},\cdots,W_{jm} \tag{9.31}$$

是 m 条线的集合，其中没有两条线相交. 在没有公共有限点的意义下，我们称这两条线平行(parallel). 证明 W_{jk} 有 m 个有限点.

557

(d)证明如果 $j\neq j'$，那么我们定义的线 W_{jk} 和 $W_{j'k'}$ 有唯一公共点.

(e)现在我们有 $m+1$ 个有 m 条平行线的集合，任意两条非平行线相交于一点. 对于(9.31)式的每一个平行线集合，现在我们加上一个不同的"无穷远点"w_j，这个点位于这个集合中的每一条线上. 设 $w_{-1}=u$ 及 $w_0=v$. 我们总共加上了 $m+1$ 个无穷远点. 于是我们再加入一条线 L，这是"无穷远线"，这条线定义为由 u，v，w_1，w_2，…，w_{m-1} 组成的线. 完成并修改我们在(b)的例子中给出的线的列表. 并列出这一例子中的所有点(有限点和无穷点).

(f)确定一般情况下构造的点的数量和线的数量.

(g)确定每一条线上的点的数量和经过每一个点的线的数量.

(h)对于所有点的集合(有限点和无穷远点)以及所有线 W_{jk} 加上无穷远线 L 的集合，验证公设 (P_1)，(P_2) 和 (P_3) 成立. [提示：首先验证 (P_2).]

参考文献

ANDERSON, I., *Combinatorial Designs: Construction Methods*, Prentice Hall, Upper Saddle River, NJ, 1990.

BETH, T., JUNGNICKEL, D., and LENZ, H., *Design Theory*, 2nd ed., Cambridge University Press, New York, 1999.

BOGART, K. P., *Introductory Combinatorics*, Pitman, Marshfield, MA, 1983.

BOSE, R. C., "On the Application of the Properties of Galois Fields to the Problem of Construction of Hyper-Graeco-Latin Squares," *Sankhyā, 3* (1938), 323–338.

BOSE, R. C., SHRIKHANDE, S. S., and PARKER, E. T., "Further Results on the Construction of Mutually Orthogonal Latin Squares and the Falsity of Euler's Conjecture," *Canad. J. Math., 12* (1960), 189–203.

BOX, G. E. P., HUNTER, W. G., and HUNTER, J. S., *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building*, Wiley, New York, 1978.

BRINKMAN, J., HODGKINSON, D. E., and HUMPHREYS, J. F., "How to Buy a Winning Ticket on the National Lottery," *Math. Gaz., 85* (2001), 202–207.

BRUCK, R. H., and RYSER, H. J., "The Nonexistence of Certain Finite Projective Planes," *Canad. J. Math, 1* (1949), 88–93.

BRUNK, M. E., and FEDERER, W. T., "Experimental Designs and Probability Sampling in Marketing Research," *J. Am. Statist. Assoc., 48* (1953), 440–452.

CHEN, K. K., BLISS, C., and ROBBINS, E. B., "The Digitalis-like Principles of *Calotropis* Compared with Other Cardiac Substances," *J. Pharmacol. Exp. Ther., 74* (1942), 223–234.

CHEN, Y., "The Steiner System $S(3,6,26)$," *J. Geometry, 2* (1972), 7–28.

CHOWLA, S., and RYSER, H. J., "Combinatorial Problems," *Canad. J. Math., 2* (1950), 93–99.

COCHRAN, W. G., and COX, G. M., *Experimental Designs*, 2nd ed., Wiley, New York, 1957.

COLBOURN, C. J., and DINITZ, J. H. (eds.), *The CRC Handbook of Combinatorial Designs*, CRC Press, Boca Raton, FL, 1996.

558

COLBOURN, C. J., DINITZ, J. H., and STINSON, D. R., "Applications of Combinatorial Designs to Communications, Cryptography, and Networking," in J. D. Lamb and D. A. Preece (eds.), *Surveys in Combinatorics, 1999*, London Mathematical Society Lecture Note Series, 267, Cambridge University Press, Cambridge, 1999, 37–100.

COX, D. R., *Planning of Experiments*, Wiley, New York, 1958.

DAVIES, H. M., "The Application of Variance Analysis to Some Problems of Petroleum Technology," Technical paper, Institute of Petroleum, London, 1945.

DIFFIE, W., and HELLMAN, M. E., "New Directions in Cryptography," *IEEE Trans. Info. Theory, 22* (1976), 644–654.

DINITZ, J. H., and STINSON, D. R. (eds.), *Contemporary Design Theory: A Collection of Surveys*, Wiley, New York, 1992.

DORFMAN, R., "The Detection of Defective Members of a Large Population," *Annals of Math. Stats., 14* (1943), 436–440.

DORNHOFF, L. L., and HOHN, F. E., *Applied Modern Algebra*, Macmillan, New York, 1978.

DU, D.-Z., and HWANG, F. K., *Combinatorial Group Testing and Its Applications*, 2nd ed., World Scientific, Singapore, 2000.

DURBIN, J. R., *Modern Algebra*, 4th ed., University of Chicago Press, Chicago, 1999.

FINNEY, D. J., *An Introduction to the Theory of Experimental Design*, University of Chicago Press, Chicago, 1960.

FISHER, J. L., *Application-Oriented Algebra*, Harper & Row, New York, 1977.

FISHER, R. A., "The Arrangement of Field Experiments," *J. Minist. Agric., 33* (1926), 503–513.

FISHER, R. A., "An Examination of the Different Possible Solutions of a Problem in Incomplete Blocks," *Ann. Eugen., 10* (1940), 52–75.

GARRETT, P., *Making, Breaking Codes: Introduction to Cryptology*, Prentice Hall, Upper Saddle River, NJ, 2001.

GILBERT, J., and GILBERT, L., *Elements of Modern Algebra*, Brooks/Cole, Pacific Grove, CA, 1999.

HALL, M., *Combinatorial Theory*, Blaisdell, Waltham, MA, 1967. (Second printing, Wiley, New York, 1980.)

HENNESSY, J. L., and PATTERSON, D. A., *Computer Architecture: A Quantitative Approach*, 2nd ed., Morgan Kaufmann Publishers, San Francisco, 1998.

HICKS, C. R., *Fundamental Concepts in the Design of Experiments*, Holt, Rinehart and Winston, New York, 1973.

HILL, R., *A First Course in Coding Theory*, Oxford University Press, New York, 1991.

HUGHES, D. R., and PIPER, F. C., *Design Theory*, 2nd ed., Cambridge University Press, New York, 1988.

JOHNSON, D. M., DULMAGE, A. L., and MENDELSOHN, N. S., "Orthomorphisms of Groups and Orthogonal Latin Squares. I," *Canad. J. Math., 13* (1961), 356–372.

JOYE, M., and QUISQUATER, J.-J., "Cryptoanalysis of RSA-Type Cryptosystems: A Visit," in R. N. Wright and P. G. Neumann (eds.), *Network Threats*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 38, American Mathematical Society, Providence, RI, 1998, 21–31.

KALISKI, B. S. (ed.), *Advances in Cryptology - CRYPTO '97. Proceedings of the 17th Annual International Cryptology Conference*, Lecture Notes in Computer Science, 1294, Springer-Verlag, Berlin, 1997.

KASKI, P., and ÖSTERGÅRD, P. R. J., "The Steiner Triple Systems of Order 19," *Math. Comp., 73*, (2004), 2075–2092.

559

KIRKMAN, T. A., "On a Problem in Combinatorics," *Camb. Dublin Math J., 2* (1847), 191–204.

KOBLITZ, N., *A Course in Number Theory and Cryptography*, 2nd ed., Springer-Verlag, New York, 1994.

LAM, C. W. H., THIEL, L. H., and SWIERCZ, S., "The Nonexistence of Finite Projective Planes of Order 10," *Canad. J. Math., 41* (1989), 1117–1123.

LI, P. C., and VAN REES, G. H. J., "Lotto Design Tables," *J. Combin. Des., 10* (2002), 335–359.

LINDNER, C. C., and RODGER, C. A., *Design Theory*, CRC Press, Boca Raton, FL, 1997.

MACNEISH, H. F., "Euler Squares," *Ann. Math., 23* (1922), 221–227.

MENEZES, A. J., VAN OORSCHOT, P. C., and VANSTONE, S. A., *Handbook of Applied Cryptography*, CRC Press Series on Discrete Mathematics and Its Applications, CRC Press, Boca Raton, FL, 1997.

PATTERSON, D. A., and HENNESSY, J. L., *Computer Organization and Design: The Hardware/Software Interface*, 2nd ed., Morgan Kaufmann Publishers, San Francisco, 1998.

PUGH, C., "The Evaluation of Detergent Performance in Domestic Dishwashing," *Appl. Statist., 2* (1953), 172–179.

RIVEST, R. L., SHAMIR, A., and ADLEMAN, L. M., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Comm. ACM, 21* (1978), 120–126.

RYSER, H. J., *Combinatorial Mathematics*, Carus Mathematical Monographs, No. 14, Mathematical Association of America, Washington, DC, 1963.

SALOMAA, A., *Public-Key Cryptography*, Texts in Theoretical Computer Science - An EATCS Series, Springer-Verlag, Berlin, 1996.

SLOANE, N. J. A., "Error-Correcting Codes and Cryptography," in D. A. Klarner (ed.), *The Mathematical Gardner*, Wadsworth, Belmont, CA, 1981, 346–382.

STINSON, D. R., "A Short Proof of the Nonexistence of a Pair of Orthogonal Latin Squares of Order Six," *J. Combin. Theory (A), 36* (1984), 373–376.

STINSON, D. R., "The Combinatorics of Authentication and Secrecy Codes," *J. Cryptology, 2* (1990), 23–49.

STINSON, D. R., *Combinatorial Designs: Constructions and Analysis*, Springer-Verlag, New York, 2003.

STREET, A. P., and STREET, D. J., *Combinatorics of Experimental Design*, Oxford University Press, New York, 1987.

TARRY, G., "Le Problème de 36 Officieurs," *C. R. Assoc. Fr. Avance. Sci. Nat., 1* (1900), 122–123.

TARRY, G., "Le Problème de 36 Officieurs," *C. R. Assoc. Fr. Avance. Sci. Nat., 2* (1901), 170–203.

WADLEY, F. M., "Experimental Design in the Comparison of Allergens on Cattle," *Biometrics, 4* (1948), 100–108.

WALLIS, W. D., *Combinatorial Designs*, Marcel Dekker, New York, 1988.

WILLIAMS, E. J., "Experimental Designs Balanced for the Estimation of Residual Effects of Treatments," *Aust. J. Sci. Res., A2* (1949), 149–168.

YATES, F., "Incomplete Randomized Blocks," *Ann. Eugen., 7* (1936), 121–140.

YOUDEN, W. J., "Use of Incomplete Block Replications in Estimating Tobacco-Mosaic Virus," *Contrib. Boyce Thompson Inst., 9* (1937), 41–48.

560

第10章 编码理论[⊖]

10.1 信息传输

本章给出编码理论的简短概述. 我们关心的是代码使用中的两方面问题：保证传输信息的隐秘性以及发现并改正传输错误. 我们讨论的方法已经应用于计算机通信、远距离空间探测器通信以及与发射台中导弹的通信，并应用于电子商务、光学和磁记录及遗传代码等方面. 今天，编码理论的发展与信息技术的进步有着密切的联系，这些技术已应用于因特网和"网络技术的新时代". 计算和电信所使用的种类繁多的网络化设备的快速发展对编码理论提出了挑战和令人振奋的新问题. 有关编码理论应用的更多内容，特别是对计算机通信方面的应用，可以参见 MacWilliams 和 Sloane[1983]. 关于代码理论的总体论述等更详细的内容，还可以参见 MacWilliams 和 Sloane[1983]、Berlekamp[1968]、Blake 和 Mullin[1975]、Cameron 和 van Lint[1991]、Goldie 和 Pinch[1991]、Hill[1986]、Peterson[1961]、Peterson 和 Weldon[1972]、PLess[1998]、Van Lint[1999]和 Welsh[1988]. 对于简洁明了的论述则可参见 Dornhoff 和 Hohn[1978]或 Fisher[1977].

信息传输中的基本步骤的模型如图 10.1 所示. 我们想象一下，从一个单词，一个英语单词或某种代码中的字，例如一个位串开始. 在步骤(a)，我们对它编码，通常是把这个字编码成一个位串. 然后在步骤(b)，我们通过一个传输信道传输这一编码字. 最后，在步骤(c)，接收到的字被解码. 这一模型适用于如电话线的物理通信路径中的传输，或跨越空间经无线电波的传输. 然而，正如 MacWilliams 和 Sloane[1983]指出的那样，类似的分析可以应用于计算机存储数据，而后又重新获取这些数据的情况，以及现代电信的其他许多数据传输中. 正如 Fisher[1977]指出的那样，这一分析也可应用于视觉到视网膜(光子的模式)上的输入. 输入被编码成视网膜中某些细胞中的电脉冲，而这些电脉冲通过神经被传输到大脑的视觉区域，在那里这些电脉冲被解码成一个视觉模式. 除些之外它还有很多其他应用.

561

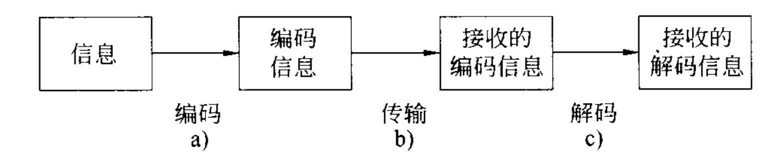


图 10.1 信息传输的基本步骤

我们将假设唯一发生错误的地方是步骤(b)，而且错误是由于噪声或弱信号引起的. 在 10.2 节，我们将讨论步骤(a)和步骤(c)，即编码和解码步骤，而不考虑传输中的错误. 10.3 节着手了解如何处理这些错误. 在 10.4 节，我们要说明如何使用编码和解码过程去发现和改正传输中的错误. 最后，10.5 节讨论使用区组设计来获得错误检测和错误校正的代码. 对于已跳过第 9 章的读者应该跳过该节.

本章着重强调的是存在问题：存在特定类型的代码吗？然后，我们还讨论优化问题：某种代码中，最好、最丰富或最大的代码是什么？

⊖ 10.1 节～10.3 节是本章的基础. 对简短论述感兴趣的读者可以直接跳到 10.5 节(而这主要依赖于第 9 章的内容).

10.2 编码与解码

有时候，图10.1的步骤(a)必定从一个包含"敏感"信息的信息产生一个编码信息. 在这种情况下，我们使用术语"加密"和"解密"来描述图10.1的步骤(a)和(c). 然而，我们将通篇使用术语"编码"和"解码". 称为*密码学*(cryptography)的领域与这样的编码和解码有关，而且如果这个代码是未知的，它还与如何破译接收到的代码有关. 这里讨论密码学的编码和解码问题，但不讨论破译问题.

要编码的信息涉及某个*信息字母表*(message alphabet)A的符号序列. 编码信息将是一个*代码字母表*(code alphabet)B的符号序列，这一字母表B可能与信息字母表A相同. 一个简单的编码规则把A的每一个符号a编码成字母表B的一个符号$E(a)$.

【例10.1 凯撒密码】 如果A和B二者都是字母表中的26个大写字母，一个简单的编码规则$E(a)$可以取$E(a)$为a后面的字母，其中$E(Z)=A$. 因此，信息

DEPOSIT SIX MILLION DOLLARS (10.1) 562

将被编码成

EFQPTJU TJY NJMMJPO EPMMBST.

如果A仍为如上定义的字母表，而$B=\{0, 1, 2, \cdots, 25\}$，那么我们可以取$E(a)$为$a$在这个字母表中的位置+2，其中我们假设Z在位置0. 这里，24+2被解释为0，而25+2被解释为1(使用9.3.1节的术语，模26的加法). 因此，$E(D)=6$. 大约在2000年前，Julius Caesar使用了类似的编码. 这些编码现在称为*凯撒密码*(Caesar cypher). ■

任何从A到B的函数$E(a)$将足以用于编码，只要我们能够适当地解码，即从$E(a)$获取a. 为了能够非歧义地做这一工作，当$a\neq b$时我们不能有$E(a)=E(b)$；即$E(a)$必须是一一对应的.

把长信息分割成符号的区组而不是单一符号，并编码区组中的信息通常很有用. 例如，如果我们使用长度为2的区组，那么信息(10.1)式变成

DE PO SI TS IX MI LL IO ND OL LA RS. (10.2)

然后，我们编码每一个2符号序列，即每一个区组[⊖]. 编码区组的一个方法是使用矩阵，如下面例子所示.

【例10.2 矩阵代码】 我们用表示字母在字母表中位置的数字来代替字母表中的每一个字母(Z在位置0). 于是，一个区组对应于一个向量. 例如，上面的区组DE对应于向量(4, 5). 假设所有区组的长度都为m. 设$\boldsymbol{M}$是一个$m\times m$矩阵，例如：

$$\boldsymbol{M}=\begin{bmatrix}2 & 3\\1 & 2\end{bmatrix}.$$

于是，我们就可以把一个区组a编码成一个区组$a\boldsymbol{M}$. 对于我们的情况，我们把区组(i, j)编码成区组$E(i, j)=(i, j)\boldsymbol{M}$. 因此，DE或(4, 5)得到如下编码：

$$(4,5)\begin{bmatrix}2 & 3\\1 & 2\end{bmatrix}=(13,22).$$

读者可以核实，当把信息(10.1)式分割成形如(10.2)式时，我们得到编码

13,22,47,78,47,75,59,98,42,75,35,57,36,60,33,57,32,50,42,69,25,38,55,92.

⊖ 为了保证信息总能被分割成相同大小的区组，我们总可以通过添加可识别作为相同字母的拷贝的"结束"串来加长信息，例如，Z或ZZZ.

一般地，我们定义的这一过程称为*矩阵编码*(matrix encoding)．它很高效．而且也容易解码．为了解码，我们把编码信息分解成各区组 b，并寻找使得 $a\boldsymbol{M}=b$ 的 a．当 $\boldsymbol{M}$ 有逆时，我们能够非歧义地解码．因为此时 $a=b\boldsymbol{M}^{-1}$．在我们的例子中，有

563

$$\boldsymbol{M}^{-1}=\begin{bmatrix}2 & -3\\ -1 & 2\end{bmatrix}.$$

例如，如果我们有编码区组(6，11)，那么我们发现它来自于下面的代码：

$$(6,11)\boldsymbol{M}^{-1}=(6,11)\begin{bmatrix}2 & -3\\ -1 & 2\end{bmatrix}=(1,4)=AD.$$

注意，如果我们有编码区组(3，4)，解码给出下面的结果：

$$(3,4)\boldsymbol{M}^{-1}=(2,-1).$$

因为(2，－1)不对应于任何字母对，所以我们可以得出在传送包含区组(3，4)的信息的过程中产生了一个错误的结论．本章的重点是如上面我们所做的那样，进行错误检测．■

一般地，假设信息被分解成长度为 k 的区组．设 A^k 是由信息字母表 A 中长度为 k 的所有区组(序列)组成的，而 B^n 是代码字母表 B 中长度为 n 的所有区组(序列)组成的．设$\mathcal{A}$是 A^k 的子集，称为*信息区组*(message block)的集合．在大多数实际应用中，$\mathcal{A}=A^k$，除非特别声明否则我们将这样假设．*区组代码*(block code)或 $k\to n$ *区组代码*($k\to n$ block code)是一个一一对应函数 E：$\mathcal{A}\to B^n$．对于$\mathcal{A}$中的所有 a，所有 $E(a)$的集合 C 定义为*代码字*(codeword)的集合．有时，称 C 为代码．在例 10.2 中，$k=n=2$，且

$$A=\{A,B,\cdots,Z\},$$
$$B=\{0,1,\cdots,25\},$$

且$\mathcal{A}=A^k$．区组(13，22)和(6，11)是代码字．然而，区组(3，4)不是代码字．在大多数实际例子中，A 和 B 都是{0，1}，信息是位串，而编码则是把长度为 k 的位串转换成长度为 n 的位串．我们将在 10.4 节看到取 $n>k$ 有助于错误检测和错误校正．

【例 10.3 重复代码】 也许编码信息的最简单方法是重复这一信息．这种类型的编码产生*重复代码*(repetition code)．假设我们如下定义 E：$A^k\to A^{pk}$：

$$E(a_1a_2\cdots a_k)=a_1a_2\cdots a_ka_1a_2\cdots a_k\cdots a_1a_2\cdots a_k,$$

其中，我们有 $a_1\ a_2\cdots a_k$ 的 p 个拷贝．例如，假设 $k=4$ 且 $p=3$．那么 $E(a_1a_2a_3a_4)=a_1a_2a_3a_4a_1a_2a_3a_4a_1a_2a_3a_4$．这是三重复代码或 $k\to 3k$ 区组代码的例子．在这样的代码中，通过比较所接收编码信息的 A^k 的连续元素很容易检测错误．甚至可以利用重复代码来校正错误．我们可以简单地使用多数解码规则．对于信息中的第 i 个数字，我们挑选 p 个拷贝中在这个位置出现最频繁的信息字母表中的字母．例如，如果 $k=4$，$p=3$，我们接收到信息 $axybauybaxvb$，那么，因为字母 x 在第二个位置上出现的次数最多，而 y 在第三个位置上出现的次数最多，我们“改正”错误，并把原来的信息解释成 $axyb$．在本章后面，我们将发现更有效的检测错误、校正错误的方法．■

564

我们通过再给出一个称为*排列代码*(permutation code)的区组代码例子来结束本节．

【例 10.4 排列代码】 假设 $A=B=\{0,1\}$，π 是{1，2，…，k}的排列．我们可以通过取 $E(a_1a_2\cdots a_k)=a_{\pi(1)}a_{\pi(2)}\cdots a_{\pi(k)}$ 来定义 E：$A^k\to B^k$．例如，假设 $k=3$，$\pi(1)=2$，$\pi(2)=3$，$\pi(3)=1$．那么 $E(a_1a_2a_3)=a_2a_3a_1$，所以 $E(011)=110$，$E(101)=011$ 等等．■

容易看到(练习 15 和练习 16)，每一个这样的排列代码和重复代码是矩阵编码．

本节练习

1. 假设

$$M=\begin{bmatrix}1 & 0\\ 2 & 4\end{bmatrix}.$$

使用由 M 定义的矩阵代码编码下面的表达式.

(a)AX (b)UV (c)BUNNZ HIL

(d)SELL ALL SHARES OF IBM (e)INVEST TWO MILLION

(f)ABORT THE MISSIONZ(注意：Z已被加入到末端，因为有奇数个字母.)

2. 假设

$$M=\begin{bmatrix}1 & 0 & 1 & 1\\ 0 & 1 & 1 & 0\end{bmatrix}.$$

利用 M，确定对应于下面每一个信息字 $a_1a_2\cdots a_k$ 的代码字 $x_1x_2\cdots x_n$.

(a)11 (b)10 (c)01 (d)00

3. 假设

$$M=\begin{bmatrix}1 & 0 & 0 & 1 & 0 & 0\\ 0 & 1 & 0 & 0 & 1 & 0\\ 0 & 0 & 1 & 0 & 0 & 1\end{bmatrix}.$$

对于下面每一个信息字重复练习 2.

(a)111 (b)101 (c)000

4. 假设

$$M=\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 1 & 1\\ 0 & 1 & 0 & 0 & 1 & 0 & 1\\ 0 & 0 & 1 & 0 & 1 & 1 & 0\\ 0 & 0 & 0 & 1 & 1 & 1 & 1\end{bmatrix}.$$

对于下面的每一个信息字重复练习 2.

(a)1111 (b)1000 (c)0001

565

5. 假设

$$M=\begin{bmatrix}1 & 2 & 3\\ 2 & 3 & 4\\ 1 & 2 & 1\end{bmatrix}.$$

使用 M 定义的矩阵代码编码下面的表达式.

(a)ABC (b)XAT (c)TTU

(d)BUY TWENTY SHARES (e)SEND THE MESSAGE AT EIGHT

(f)OBSERVE THE TRANSFERSZZ(注意：已加入两个Z使得字母的数量可以被3整除.)

6. 对于下面的矩阵 M 重复练习 1：

$$M=\begin{bmatrix}4 & 6\\ 1 & 2\end{bmatrix}.$$

7. 对于下面的矩阵 M 重复练习 5：

$$M=\begin{bmatrix}1 & 1 & 0\\ 1 & 2 & 4\\ 2 & 1 & 0\end{bmatrix}.$$

8. 假设我们在矩阵代码中使用练习 1 中的 M. 解码下面的代码或证明产生一个错误，即这不是正确的代码.

(a)(5, 8) (b)(8, 12) (c)(1, 8) (d)(7, 12)

(e)16, 48, 44, 60 (f)8, 16, 8, 12 (g)6, 10, 51, 69, 20, 20

9. 假设我们把练习 5 中的矩阵 M 用于矩阵代码中. 解码下面的代码或证明产生一个错误，即这不是正确的

代码.

(a)(8, 14, 14)　　(b)(17, 35, 23)　　(c)(24, 47, 30)

(d)(52, 95, 102, 46, 72, 88)　　(e)(48, 91, 70, 15, 27, 23)

10. 对于练习 6 中的矩阵 $\boldsymbol{M}$ 重复练习 8.

11. 对于练习 7 中的矩阵 $\boldsymbol{M}$ 重复练习 8.

12. 对于下面$\{1, 2, \cdots, k\}$的每一个排列，确定在下面对应的排列代码中表示的 E 的值.

(a)$k=3$, $\pi(1)=3$, $\pi(2)=2$, $\pi(3)=1$.

i. $E(110)$　　ii. $E(010)$　　iii. $E(011)$

(b)$k=4$, $\pi(1)=3$, $\pi(2)=4$, $\pi(3)=1$, $\pi(4)=2$.

i. $E(0110)$　　ii. $E(1100)$　　iii. $E(0111)$

(c)$k=5$, $\pi(1)=5$, $\pi(2)=1$, $\pi(3)=4$, $\pi(4)=2$, $\pi(5)=3$.

i. $E(00001)$　　ii. $E(10101)$　　iii. $E(11010)$

13. 如果A是由区组 010, 111, 001 和 110 组成的，且 $E(abc)$是 $a+b$, $a+c$, $b+c$, 其中加法是模 2 加法，那么确定 C.

14. 构建一个 $1\to2$ 区组代码的例子.

15. 证明每一个排列代码是一个矩阵编码.

566 16. 证明每一个重复代码是一个矩阵编码.

10.3 错误校正码

10.3.1 错误校正和汉明距离

本节研究传输中检测错误和校正错误的代码使用. 特别地，我们研究图 10.1 的步骤(b). 我们假设从一个已编码信息开始，这一编码信息是使用区组代码编码的，代码字母表中的区组的长度为 n. 举一个具体的例子，我们假设编码信息是一个位串，所以所有的区组是长度为 n 的位串. 因此，我们讨论的是*二进制代码*(binary code)，或*二进制区组代码*(binary block code)，或*二进制 n 代码*(binary n-code). 在练习 7 中，我们修改这一假设，且取编码信息为数字表$\{0, 1, \cdots, q-1\}$的位串. 于是，我们讨论的是 *q 进制代码*(q-ary code). 通过区组输送信息. 我们假设在编码中不存在错误，所以被发送的区组仅是代码字. 我们假设在传输过程中唯一可能发生的错误是数字 0 和 1 的互换. 对其他错误，如漏掉一个数字或多出一个数字都不加以考虑. 这些错误很容易检测，因为所有区组都有相同的长度. 根据习惯作法，我们假设把 1 交换成 0 的概率与把 0 交换成 1 的概率相同，而且每一个数字处发生错误的概率相同，且与前面可能出现的错误无关[⊖]. 在这种情况下，我们讨论*二进制对称信道*(binary symmetric channel). 关于这些假设的含义，请参见 10.3.3 节.

回想一下，我们只发送代码字. 如果我们接收到一个不是代码字的区组，那么我们检测到了一个传输错误. 例如，如果发送代码字 10010，而接收到的是区组 10110，而且如果 10110 不是代码字，那么我们知道存在一个错误. 存在这样的情况，我们希望校正这个错误，即猜测所发送的代码字是什么，特别是当我们不能要求重新发送的时候. 例如，在从空间探测器(例如水手 9 号火星探测器，下面我们讨论其代码)传输照片，或传输依赖于老式磁带时即使如此. 错误检测和错误校正码有很多应用. 例如，称为奇偶校验代码的简单错误检测代码(参见例 10.6)被用于最早的计算机上：UNIVAC，旋风 I 和 IBM650(Wakerly[1978]). 实际上，正是编码方法可以校正贝尔实验

⊖ 非随机发生的错误有时会突发性地发生，几处错误在一起连续出现. 关于突发错误校正码的更多内容，参见 Bossert 等[1997].

室所使用的早期计算机中的错误的想法，导致 Hamming[1950]开发错误校正码. 如今，这样的代码广泛用于容错计算机的设计中(关于错误检测和错误校正码在计算机中的应用，可以参见 MacWilliams 和 Sloane[1983]、Pless[1998]、Poli 和 Huguet[1992]、Sellers，Hsiao 和 Bearnson[1968]、Wakerly[1978]，以及由 Avizenius[1976]、Carter 和 Bouricius[1971]所写的研究文章). 这样的代码也是光盘设计的基础，而光盘带来了音乐产业革命(正如我们在 10.5 节中所注意到的那样).

现在有很多设计错误校正码的好方法. 假设我们选择代码，使得在代码字集合中，没有两个代码字太接近或太相似. 例如，假设仅有如下的代码字：

567

$$000000,\quad 010101,\quad 101010,\quad 111111, \tag{10.3}$$

那么我们几乎不可能把其中的两个混淆，而接收到的不同于代码字的信息可以容易地解释成最靠近它的代码字.

让我们使最接近的概念更加明确. 相同长度的两个位串之间的*汉明距离*(Hamming distance)是它们的相应位中数字不同的位的数量[⊖]. 因此，如果使用 $d(\cdot,\ \cdot)$来表示这一距离，我们有

$$d(000000,010101)=3.$$

我们的目标是寻找在汉明距离的意义下没有两个字太接近的代码字集合.

假设我们已找到一个代码字集合(根据我们现在的假设，所有代码字的长度相同). 假设这些代码字之间的最小距离是 d. 那么我们可以检测所有 $d-1$ 或更少个数字的错误. 因为，如果 $d-1$ 或更少个数字被交换，那么结果位串将不是代码字，而且我们能够识别或检测出存在一个错误. 例如，如果可能的代码字就是(10.3)式中的那些代码字，那么 d 等于 3，而且我们能够检测出最多两个数字的错误. 假设我们使用这样的策略，即如果我们接收到一个非代码字，那么我们把它解释成在汉明距离的意义下最接近的代码字(在距离相同时随机选取). 这一规则称为*最近邻居规则*(nearest-neighbor rule). 例如，如果代码字是(10.3)式中的代码字，而且我们接收到了字 010000，我们将把这个字解释成 000000，因为

$$d(010000,000000)=1,$$

而对于其他代码字 α，都有

$$d(010000,\alpha)>1.$$

使用最近邻居规则，我们可以校正涉及小于 $d/2$ 个数字的所有错误. 因为如果小于 $d/2$ 个数字被交换，那么其结果位串最接近于(被传输的)正确的代码字，根据最近邻居规则，它被解释成这个代码字. 因此，我们有*错误校正码*(error-correcting code). 能校正至多 t 个错误的代码称为 *t 错误校正码*(t-error-correcting code).

我们把这些结果概括如下.

定理 10.1 *假设 d 是二进制代码 C 的两个代码字之间的最小(汉明)距离. 那么代码 C 可以检测出至多 $d-1$ 个错误，使用最近邻居规则，可以校正至多$\lceil (d/2)-1\rceil$个错误.*

下一个定理说的是没有哪一个错误校正规则比最近邻居规则做得更好.

568

定理 10.2 *假设 d 是二进制代码 C 的两个代码字之间的最小(汉明)距离. 那么没有可以检测出多于 $d-1$ 个错误的错误校正规则，也没有可以校正多于 $t=\lceil (d/2)-1\rceil$个错误的错误校正规则.*

证明 因为 d 是最小(汉明)距离，所以存在两个代码字 α 和 β 有 $d(\alpha,\ \beta)=d$. 如果 α 被传送，且有 d 个错误发生，那么可能接收到 β. 因此，检测不出错误. 接下来，注意，错误校正规

⊖ 汉明距离是以 R. W. Hamming 的名字而命名的，他写了关于错误检测和错误校正码的最早的文章(Hamming[1950]).

则给每一代码字 λ(即每一个长度为 n 的位串 λ)指定一个代码字 $R(\lambda)$. 设 γ 是与 α 和 β 不同的词且有 $d(\alpha, \gamma)\leqslant t+1$, $d(\beta, \gamma)\leqslant t+1$. 那么 $R(\gamma)$一定是某个代码字. 这个代码字不能是 α 也不能是 β. 不失一般性，比如说它不是 α. 那么传送 α，且至多有 $t+1$ 个错误接收到 γ. 这个接收字被校正为不同于 α 的 $R(\gamma)$. ■

因为最近邻居规则(以及任意其他合理的规则)不允许我们校正所能检测到的所有错误，所以有时候，我们不打算校正错误，而只是检测错误，如果检测到错误就要求重新传输. 在发送发射导弹或改变空间发射路线命令这样的重要场合，就是这样的情形.

假设我们知道什么样的错误可能发生. 如果经常发生这样的错误，那么我们希望代码字具有相当大的最小汉明距离. 如果错误不常发生，那么最小汉明距离可以更小，例如，如果发生的错误数量不超过 1，那么这个距离可以是 3. 一般地，我们希望能够构造出有给定长度 n 和最小给定汉明距离 d 的代码字集合. 这样的二进制代码的代码字集合将称为(n, d)代码((n, d)-code).

如果 $d\leqslant n$，那么总存在(n, d)代码. 设这个代码字集合是

$$\underbrace{000\cdots0}_{n\text{个}0} \quad \text{和} \quad \underbrace{111\cdots1}_{d\text{个}1}\underbrace{00\cdots0}_{n-d\text{个}0}.$$

这一代码的麻烦是代码的数量很少. 我们只能编码两个不同信息区组的集合 A. 我们想要构建更丰富代码的方法，即有更多可能代码字的代码.

在 10.4 节，我们将看到如何使用更聪明的编码方法 $E: A\rightarrow B^n$ 来寻找更丰富的(n, d)代码. 在 10.5 节，我们将看到如何利用区组设计，特别是它们的关联矩阵来定义非常丰富的(n, d)代码. 在下一小节，我们得到(n, d)代码大小的上界.

在例 10.3 中，我们介绍了重复代码的概念.

【例 10.5 重复代码(再探例 10.3)】 正如我们在例 10.3 中所描述的那样，重复一个区组 r 次的重复代码 $k\rightarrow rk$ 代码取 $E(a)=aa\cdots a$，其中 a 被重复 r 次. 使用这样的代码，我们能够容易地检测错误及校正错误. 例如，考虑三重复代码的情况. 代码字集合 C 是由长度为 $3k$ 的区组组成的，这样的区组是长度为 k 的区组被重复三次. 为了检测错误，我们在 k 区组的三个重复中比较第 i 个数字. 因此，假设 $k=4$ 且我们接收到 001101110110，并把这个接收字分解成 k 区组 0011/0111/0110. 此时，我们知道存在一个错误，例如，前两个 k 区组的第二个数字不同. 我们用这样的方法可以检测到至多两个传输错误. 然而，如果错误都出现在原来 k 区组中的相同数字上，那么也许有三个错误没有被检测到. 当然，两个代码字 aaa 之间的最小距离是 $d=3$，所以这一观察与定理 10.2 的结果一致. 我们可以尝试如下校正这一错误：注意 k 区组的第二个数字是 1 两次而 0 一次. 如果我们假设错误不常发生，那么可以使用多数规则并假设正确数字是 1. 使用类似的推理，我们可以解码 001101110110 为 0111. 这种错误校正过程能校正至多一个错误. 然而，如果两个错误出现在相同的位置上，那么它们被不正确地"校正". 这一观察再一次与定理 10.2 的结果一致，因为我们有 $d=3$ 的代码. 如果我们希望校正更多的错误，我们可以简单地使用更多重复. 例如，5 重复代码 $E(a)=aaaaa$ 有 $d=5$，这一代码可以检测至多 4 个错误，校正至多 2 个错误. 遗憾的是，设计错误校正代码的重复方法在时间和空间上是昂贵的. ■

569

【例 10.6 奇偶校验码】 检测单一错误的一个简单方法是在一个区组中加入一个数字，这个数字总是把 1 的数量变成偶数. 所加入的这个额外数字称为奇偶校验数字(parity check digit). 对应的代码是 $k\rightarrow(k+1)$代码，它让 $E(0011)=00110$，$E(0010)=00101$ 等等. 我们可以通过下式表示 E:

$$E(a_1a_2\cdots a_k) = a_1a_2\cdots a_k\sum_{i=1}^{k}a_i,$$

其中 $\sum_{i=1}^{k} a_i$ 被解释成模2求和，这与9.3.1节的定义一致. 贯穿本章的其余部分，加法都解释成模加法. 因此，$1+1=0$，这就是为什么 $E(0011)=00110$. 在这样的奇偶校验码(parity check code) E 中，两个代码字的最小距离是 $d=2$，所以可以检测出一个错误. 没有错误可以得到校正. 正如我们前面所指出的那样，这样的奇偶校验码被用于最早的计算机中：UNIVAC，旋风 I 和 IBM650. 20世纪60年代的各种大型和小型计算机都有内存奇偶校验作为附加功能，到了20世纪70年代，这一校验已成为很多机器的标准. ■

10.3.2 汉明界

在(n, d)代码 C 中固定给定位串 s. 设长度为 n 的距位串 s 距离正好为 t 的位串的集合被记为 $B_t(s)$. 于是 $B_t(s)$中的位串数量等于$\binom{n}{t}$，因为我们从 n 个位置中选出 t 个位置改变其中的数字. 设$B_t'(s)$表示距 s 的距离至多为 t 的长度为 n 的位串集合. 因此，$B_t'(s)$中的位串数量为

570

$$b=\binom{n}{0}+\binom{n}{1}+\cdots+\binom{n}{t}. \tag{10.4}$$

如果 $t=\lceil (d/2)-1\rceil$，那么对于 C 中每一个 $s\neq s^*$，有 $B_t'(s)\cap B_t'(s^*)=\varnothing$. 因此，对于 $t=\lceil (d/2)-1\rceil$，长度为 n 的每一个位串至多在一个集合 $B_t'(s)$中. 因为长度为 n 的位串有 2^n 个，所以我们有

$$|C|b=\sum_{s\in C}|B'_t(s)|=|\bigcup_{s\in C}B'_t(s)|\leqslant 2^n.$$

因此，我们有下面的定理.

定理 10.3(汉明界) 如果 C 是(n, d)代码且 $t=\lceil (d/2)-1\rceil$，那么

$$|C|\leqslant\frac{2^n}{\binom{n}{0}+\binom{n}{1}+\cdots+\binom{n}{t}}. \tag{10.5}$$

这一结果是 Hamming[1950]提出的. 有时我们称其为球封装边界(sphere packing bound).

为了说明这一定理，我们取 $n=d=3$. 那么 $t=1$. 我们发现，任意的(n, d)代码至多有

$$\frac{2^3}{\binom{3}{0}+\binom{3}{1}}=2$$

个码字. 存在两个代码字的(3，3)代码，即位串111和000. 我们将在10.5.3节中研究(n, d)代码大小的另一个上界.

10.3.3 错误的概率

回想一下，我们假设有一个二进制对称信道：从1切换到0的概率与从0切换到1的概率相同，而这个共同概率 p 对每一个数字都相同，与任意前面可能已发生的错误无关. 这时，我们有下面的定理.

定理 10.4 在二进制对称信道中，传输长度为 n 的位串正好产生 r 个错误的概率由下式给出：

$$\binom{n}{r}p^r(1-p)^{n-r}.$$

证明[⊖] 熟悉概率知识的读者会认识到我们现在所处的情况与例5.35中所描述的伯努利试验的情况一样. 我们寻找任何次试验的成功概率为 p 时 n 个独立的重复试验中成功 r 次的概率.

571

⊖ 对概率不熟的读者可以跳过这一证明.

定理中所给的公式就是这一概率的著名公式(参见例 5.35). ■

例如，假设 $p=0.01$. 那么在一个 4 数字信息中，没有错误的概率是

$$(1-0.01)^4=0.960\,596,$$

有一个错误的概率是

$$\binom{4}{1}(0.01)(1-0.01)^3=0.038\,812,$$

有两个错误的概率是

$$\binom{4}{1}(0.01)^2(1-0.01)^2=0.000\,588,$$

有多于两个错误的概率是

$$1-0.960\,596-0.038\,812-0.000\,588=0.000\,004.$$

比较例 10.5 和例 10.6 中的两个代码是有益的. 假设我们希望发送一条有 1000 个数字的信息，且错误的概率为 $p=0.01$. 如果不进行编码，那么不出现错误的概率等于$(1-0.01)^{1000}$，这个概率近似等于 0.000 043. 假设作为对照，我们使用例 10.5 的 $k\to 3k$ 三重复代码，且 $k=1$. 假设我们想要发送一个数字 a. 我们把它编码为 aaa. 正确地发送区组 aaa 的概率等于$(0.99)^3$，或近似等于 0.970 299. 根据定理 10.4，一个错误的概率是 $\binom{3}{1}(0.01)(0.99)^2$，或近似于 0.029 403. 因此，因为我们能够校正一个错误，正确地解释信息 aaa 的概率和正确地解码单一数字 a 的概率为 $0.970\,299+0.029\,403=0.999\,702$. 在原来的信息中有 1000 个数字，所以正确地解码整个信息的概率是$(0.999\,702)^{1000}$，或近似于 0.742 268. 这个概率要比 0.000 043 大的多. 注意，大幅度增加无错误传输可能性需要如下的代价：为了接收到 1000 个数字的信息，我们需要传送 3000 个数字.

让我们与奇偶校验码做一下比较. 假设我们把这 1000 数字信息分解成长度为 10 的区组；总共有 100 个区组. 每一个区组通过加入一个奇偶校验数字被编码成 11 数字区组. 发送 11 数字区组而不产生任何错误的概率是$(0.99)^{11}$，或近似等于 0.895 338. 同样，根据定理 10.4，发送这个区组只产生一个错误的概率是$\binom{11}{1}(0.01)(0.99)^{10}$，或近似等于 0.099 482. 现在，如果产生一个错误，那么我们能够检测出它并要求重新发送. 因此，可以合理地假设我们可以消除单一错误. 因此，正确地接收到 11 数字区组的概率为

572

$$0.895\,338+0.099\,482=0.994\,820.$$

现在，原来的 1000 数字信息有 100 个长度为 10 的区组，所以最后正确地解码整个信息的概率为 $0.994\,820^{100}$，或近似等于 0.594 909. 正确传输的概率小于使用三重复代码的概率，但是远大于不编码的传输概率. 其代价远小于三重复代码：为得到 1000 个数字，我们需要发送 1100 个数字.

10.3.4 合意解码及其与寻找分子序列中的模式之间的关系⊖

考虑在一个很容易产生错误的“吵杂”的二进制对称信道内传输的情况. 如重复代码的情况那样，在接收 B_n 中的位串集合，例如位串 x_1，x_2，…，x_p 时，我们也许要求重新发送若干次. 这些位串中的某些在代码字集合 C 中，而另一些不在 C 中. 我们希望确定 C 中的哪些代码字是预期的代码字. 这个问题是在 C 中寻找与实际接收的字在某种意义上是“合意”的字. “多数规则解码”的想法是这一情况的特殊情况. 这一合意问题在很多应用中都会遇到，包括投票和“群策”，

⊖ 本小节可以跳过.

以及从分子序列数据库中选取最接近的匹配，或因特网上(比较若干搜索引擎的结果)的“元搜索”等等．在有关“数学社会学”的文献中人们对合意问题的数学做了广泛的研究．有关这一课题的介绍，请参见 Bock[1988]、Johnson[1998]和 Roberts[1976].

一种广泛使用的合意过程是*中值过程*(median procedure)：寻找 C 中 $\sum_{i=1}^{p} d(w,x_i)$ 最小的代码字 w，其中 d 是汉明距离．另一个合意过程，*平均值过程*(mean procedure)，是在 C 中寻找 $\sum_{i=1}^{p} d(w,x_i)^2$ 最小的代码字 w. 例如，考虑由(10.3)式中的代码字组成的代码字集合 C. 假设我们要求传输信息 3 次并接收到如下 3 个字：

$$x_1 = 100000,\quad x_2 = 110000,\quad x_3 = 111000.$$

我们计算

$$\sum_{i=1}^{3} d(000000,x_i) = 1+2+3 = 6,\quad \sum_{i=1}^{3} d(010101,x_i) = 4+3+4 = 11,$$

$$\sum_{i=1}^{3} d(101010,x_i) = 2+3+2 = 7,\quad \sum_{i=1}^{3} d(111111,x_i) = 5+4+3 = 12.$$

因为 000000 有最小和，所以中值过程选出它．我们称它为*中值*(median)．类似地，计算　573

$$\sum_{i=1}^{3} d(000000,x_i)^2 = 1+4+9 = 14,\quad \sum_{i=1}^{3} d(010101,x_i)^2 = 16+9+16 = 41,$$

$$\sum_{i=1}^{3} d(101010,x_i)^2 = 4+9+4 = 17,\quad \sum_{i=1}^{3} d(111111,x_i)^2 = 25+16+9 = 50,$$

所以，平均过程同样选出代码字 000000. 我们称它为*平均值*(mean).

中值过程和平均值过程的结果不总相同，另外，它们也不总是给出唯一的解码，即可能存在歧义性．例如，考虑代码字集合

$$C = \{111111, 001110\}.$$

假设接收到字 $x_1=001111$，$x_2=101011$. 那么

$$\sum_{i=1}^{2} d(111111,x_i) = 2+2 = 4,\quad \sum_{i=1}^{2} d(001110,x_i) = 1+3 = 4,$$

$$\sum_{i=1}^{2} d(111111,x_i)^2 = 4+4 = 8,\quad \sum_{i=1}^{2} d(001110,x_i)^2 = 1+9 = 10.$$

中值过程导致歧义性，给出 C 中的两个代码字为中值．而平均过程导致唯一解 111111 为平均值.

在社会学和生物学的很多问题中，数据以某个字母表 Σ[⊖] 上的序列或“字”的形式出现．给定序列的集合，我们寻找一个广泛出现的*模式*(pattern)或*特性*(feature)，我们把这一模式看成*合意序列*(consensus sequence)或序列的集合．一个模式通常被认为是一个短的且有固定长度的连续子序列．在例 6.11，我们已注意到，在分子序列中这种模式，或合意序列的发现已导致了许多重要的发现，例如，促进身体生长的血小板衍生因子的序列有 87%的部分等同于引发癌症的基因 v-sis 序列．这导致了人们发现 v-sis 是通过刺激生长产生影响的．为了评估模式与一个序列的吻合程度，我们必须度量不同长度的字之间的距离．如果 b 的长度大于 a，那么作为 b 的连续子

⊖ 下面所进行的分子序列中的模式识别的讨论来自于 Mirkin Roberts[1993]. 这是称为*生物合意*(bioconsensus)领域中的一个例子，这一领域包括基于社会学方法在生物学领域的应用．生物合意的诸多典型问题中，使用不同的方法或模型产生了若干选择方案(诸如种系发生树筛选或分子序列筛选)，我们需要寻找合意解．Day 和 Mcmoriis[1992]概述了合意方法在分子生物学中的使用．Day[2002]和 Day 及 McMorris[1993]给出了很多参考资料，在“对准与合意”领域有上百篇论文．例如，Kannan[1995]概述了种系发生树重构的合意方法．有关更多的信息，请参见 Day 和 McMorris[2003]和 Janowitz 等[2003].

序列，$d(a, b)$可以是a的所有可能的对准中错配的最小数目．我们称这个最小数目为**最佳错配**

[574] **距离**(best-mismatch-distance)．如果这些序列是位串，这个最佳错配距离就是与a等长度的b的所有连续子序列与a之间的最小汉明距离．例如，考虑$a=0011$，$b=111010$．那么可能的对准是

```
111010      111010      111010
0011          0011          0011.
```

最佳错配距离是2，在第三个对准中达到了这个距离．度量$d(a, b)$的另一个方法是计数序列间的最小错配数，这些序列可以通过在a和b的适当位置插入空格而得到(其中，Σ的字母与空格间的错配作为通常的错配计数)．尽管这个方法在分子生物学中有着广泛的应用，但我们不使用这种距离度量方法．

Waterman[1989]和其他文章(Waterman，Arratia 和 Galas[1984]、Galas，Eggert 和 Waterman[1985]、Waterman[1995])研究了这样的情况，其中Σ是一个有限的字母表，M是一个固定的有限数(模式的长度)，$\mathcal{X}=\{x_1, x_2, \cdots, x_n\}$是$\Sigma$的长度为$L$的字(序列)的集合，其中$L \geqslant M$，我们从$\Sigma$寻找长度为$M$的合意字的集合$F(\mathcal{X})=F(x_1, x_2, \cdots, x_n)$．这个合意字集合可能是$\Sigma^M$中的任意$n$个字，或者认可模式字的固定子集中的字．从现在起，我们将不考虑后者．这里，取自于 Waterman[1989]的数据小片段，在该论文中，他考察了59个细菌促长因子序列：

RRNABP1	ACTCCCTATAATGCGCCA
TNAA	GAGTGTAATAATGTAGCC
UVRBP2	TTATCCAGTATAATTTGT
SFC	AAGCGGTGTTATAATGCC.

注意，如果我们要寻找长度为4的模式，那么每一个序列有模式TAAT．然而，假设我们加入另一个序列：

M1RNA　　AACCCTCTATACTGCGCG.

模式TAAT不在这里出现．然而，因为字TACT出现，所以它几乎出现，而且这个字与模式TAAT只有一个错配．所有，在某种意义上，模式TAAT是一个良好的合意模式．现在，我们使这一想法更精确．

实际上，问题要比我们刚才所描述的更复杂．我们有长序列，而且我们考虑开始于固定位置，比如说第j个位置的长度为L的"窗口"．因此，我们考虑在一个长序列中开始于第j个位置的长度为L的字．对于每一个长度为M的可能模式，我们要问在开始于第j个位置的长度为L的窗口内的每一个序列与它匹配的程度．为了公式化这一描述，设Σ是一个大小至少为2的有限字母表，而且X是Σ上长度为L的字的有限集合．设$F(\mathcal{X})$是长度为$M \geqslant 2$的字的集合，也就是我们的**合意模式**(consensus pattern)．设$\chi=\{x_1, x_2, \cdots, x_p\}$．定义$F(\mathcal{X})$的一种方法如下所

[575] 述．设$d(a, b)$是最佳错配距离．考虑非负参数λ_d，这一参数伴随着d单调递减，且设$F(x_1, x_2, \cdots, x_p)$是Σ^M的长度为M且使下式最大的字ω的全体：

$$s_\chi(w) = \sum_{i=1}^{p} \lambda_{d(w, x_i)}.$$

我们称这样的F为 Waternman 合意(Waterman consensus)．特别地，Waterman 和其他人使用参数$\lambda_d=(M-d)/M$.

作为例子，我们注意到一个频繁使用的字母表是嘌呤/嘧啶字母表$\{R, Y\}$，其中$R=A$或G，$Y=C$或T．为简便起见，使用数字0，1而不是字母R，Y更容易．因此，设$\Sigma=\{0, 1\}$，$M=2$，并考虑$F(x_1, x_2)$，其中$x_1=111010$，$x_2=111111$．可能的模式字是00，01，10，11．我们有

$$d(00,x_1)=1,\quad d(00,x_2)=2,$$
$$d(01,x_1)=0,\quad d(01,x_2)=1,$$
$$d(10,x_1)=0,\quad d(10,x_2)=1,$$
$$d(11,x_1)=0,\quad d(11,x_2)=0.$$

因此

$$s_\chi(00)=\sum_{i=1}^{2}\lambda_{d(00,x_i)}=\lambda_1+\lambda_2,$$
$$s_\chi(01)=\sum_{i=1}^{2}\lambda_{d(01,x_i)}=\lambda_0+\lambda_1,$$
$$s_\chi(10)=\sum_{i=1}^{2}\lambda_{d(10,x_i)}=\lambda_0+\lambda_1,$$
$$s_\chi(11)=\sum_{i=1}^{2}\lambda_{d(11,x_i)}=\lambda_0+\lambda_0.$$

根据 Waterman 合意的定义，只要 $\lambda_0>\lambda_1>\lambda_2$，那么 11 就是合意模式.

作为另一个例子，设 $\Sigma=\{0,1\}$，$M=3$，考虑 $F(x_1, x_2, x_3)$，其中 $x_1=000000$，$x_2=100000$，$x_3=111110$. 可能的模式字是 000，001，010，011，100，101，110，111. 我们有

$$s_\chi(000)=\lambda_2+2\lambda_0,\quad s_\chi(001)=\lambda_2+2\lambda_1,\quad s_\chi(100)=2\lambda_1+\lambda_0.$$

现在，$\lambda_0>\lambda_1>\lambda_2$ 意味着 $s_\chi(000)>s_\chi(001)$. 类似地，可以证明，$s_\chi(000)$或 $s_\chi(001)$最大化 s_χ. 单调性不能说明这些中哪一个更大.

另一个合意过程是使用中值过程的一个变形，这个过程给出最小化下式的所有长度为 M 的字 w:

576

$$\sigma_\chi(w)=\sum_{i=1}^{p}d(w,x_i);$$

或者我们可以使用平均过程的一个变形，这个过程给出最小化下式的长度为 M 的所有字.

$$T_\chi(w)=\sum_{i=1}^{p}d(w,x_i)^2.$$

例如，假设 $\Sigma=\{0,1\}$，$M=2$，$\chi=\{x_1, x_2, x_3, x_4\}$，$x_1=1111$，$x_2=0000$，$x_3=1000$，$x_4=0001$. 那么可能的模式字是 00，01，10，11. 我们有

$$\sum_{i=1}^{4}d(00,x_i)=2,\quad \sum_{i=1}^{4}d(01,x_i)=3,$$
$$\sum_{i=1}^{4}d(10,x_i)=3,\quad \sum_{i=1}^{4}d(11,x_i)=4.$$

因此，00 是中值. 然而，我们有

$$\sum_{i=1}^{4}d(00,x_i)^2=4,\quad \sum_{i=1}^{4}d(01,x_i)^2=3,$$
$$\sum_{i=1}^{4}d(10,x_i)^2=3,\quad \sum_{i=1}^{4}d(11,x_i)^2=6,$$

所以平均过程产生两个字 01 和 10，其中任何一个都不是中值.

现在，我们考虑带有特殊参数 $\lambda_d=(M-d)/M$ 的 Waterman 合意，这个参数常用于实际应用中. 回想一下，我们有

$$\sigma_\chi(w)=\sum_{i=1}^{p}d(w,x_i),$$

$$s_{\chi}(w)=\sum_{i=1}^{p}\lambda_{d(w,x_i)}=p-\frac{1}{M}\sum_{i=1}^{p}d(w,x_i).$$

因此，对于固定的$M\geqslant 2$，大小至少为2的Σ，以及长度为$L\geqslant M$的x_i的任意大小的集合$\mathcal{X}$，对于所有长度为M的字w和w'，有

$$\sigma_{\chi}(w)\geqslant\sigma_{\chi}(w')\leftrightarrow s_{\chi}(w)\leqslant s_{\chi}(w').$$

因此，对于固定的$M\geqslant 2$，大小至少为2的Σ，以及长度为$L\geqslant M$的xi的任意大小的集合$\mathcal{X}$，存在使得Waterman合意与中值相同的参数λ_d的选择．（对于$M=1$或$|\Sigma|=1$，这一结论也成立，但是此时是平凡的情况．）

类似地，可以证明(见练习32和练习33)，对于固定的$M\geqslant 2$，大小至少为2的Σ，以及长度为$L\geqslant M$的x_i的任意大小的集合$\mathcal{X}$，存在参数λ_d的选择使得对于长度为M的所有字w和w'，有

$$\tau_{\chi}(w)\geqslant\tau_{\chi}(w')\leftrightarrow s_{\chi}(w)\leqslant s_{\chi}(w').$$

对于这样的λ_d的选择，一个字是Waterman合意的当且仅当它是平均值．令人感到惊讶的是，广泛使用的Waterman合意实际上可以是不易识别的中值或平均值．

577

本节练习

1. 在下列每一种情况中，确定两个代码字x和y之间的汉明距离．

   (a)$x=1010001$，$y=0101010$　(b)$x=11110011000$，$y=11001001001$

   (c)$x=10011001$，$y=10111101$　(d)$x=111010111010$，$y=101110111011$

2. 在奇偶校验码中，对于下列各$\boldsymbol{a}$，确定$E(\boldsymbol{a})$．

   (a)111111　(b)1001011

   (c)001001001　(d)01010110111

3. 对于下列每一种代码C，确定可能被检测出的错误数量，以及使用最近邻居规则所能校正的错误数量．

   (a)$C=\{00000000,\ 11111110,\ 10101000,\ 01010100\}$

   (b)$C=\{000000000,\ 111111111,\ 111110000,\ 000001111\}$

   (c)$C=\{000000000,\ 111111111,\ 111100000,\ 000011111,\ 101010101,\ 010101010\}$

4. 对于下面每一种二进制代码C和接收串x_1，x_2，…，x_p，求所有中值和平均值．

   (a)C：{00000000，11111110，10101000，01010100}

   x_i：00011000，01000010，01100110

   (b)C：{000000000，111111111，111110000，000001111}

   x_i：000101000，010010010，011010110

   (c)C：{000000000，111111111，111100000，000011111，101010101，010101010}

   x_i：100011001，011000110，000110101，110000011，010001010

5. 对于下列各组a和b，确定最佳错配距离$d(a,\ b)$．

   (a)$a=01$，$b=1010$　(b)$a=001$，$b=101010101010101$

   (c)$a=00$，$b=1101011011$　(d)$a=101$，$b=100110$

6. 证明存在长度为n的代码字集合C和长度为n的信息集合，使得中值过程给出唯一解而平均值过程不给出唯一解．

7. q进制区组n代码(q-ary block n-code)(q进制代码(q-ary code))是从数字表{0，1，…，$q-1$}选出的长度为n的串集合．这个数字表上的两个串之间的汉明距离(Hamming distance)同样定义为不同数字的数目．例如，如果$q=4$，那么$d(0123,\ 1111)=3$．

   (a)确定下列串x和y之间的汉明距离．

   i. $x=0226215$，$y=2026125$

ii. x=000111222333，y=001110223332

iii. x=01010101010，y-01020304050

(b)如果q进制区组n代码中两个代码字之间的最小距离是d，那么这个代码能检测出多少个错误？

(c)在最邻近规则下，这个代码可以校正多少个错误？ 578

8. 对于每一个代码字的长度为10，且任意两个代码字之间的最小距离为7的代码，确定代码字数量的上界.
9. 对于代码字的长度为11，且两个代码字之间的最小距离为6的情况，重复练习8.
10. 对于代码字的长度为7，且可以校正至多2个错误的代码c，确定代码字数量的上界.
11. 假设代码字是长度为10的位串，我们有二进制对称信道，且错误的概率是0.1. 确定在发送一个代码字时，产生下面错误的概率：

    (a)没有错误　　(b)正好只有一个错误

    (c)正好有两个错误　　(d)多于两个错误
12. 如果代码字的长度是6而错误概率是0.001，重复练习11.
13. 假定我们假设在传输一个q进制代码字(练习7)时，错误的概率p对于每一个数字都是相同的，且与任意前面的错误无关. 另外，如果在第i个数字上有一个错误，那么这个数字被转变成其余$q-1$个符号中的任意一个符号的概率均等. 于是我们称这是一个*q进制对称信道*(q-ary symmetric channel).

    (a)在一个q进制对称信道中，当发送长度为n的串时，正好有t个错误的概率是多少？

    (b)当$q=3$时，发送11111而接收到22222的概率是多少？

    (c)当$q=4$时，发送1111而接收到2013的概率是多少？
14. 假设在一个二进制对称信道内，代码字的长度为$n=3$且错误概率为0.1. 如果我们想要这样的代码：如果接收信息是错误的，那么这个代码校正这个接收信息的概率是$\lambda\geqslant 0.95$，这时两个代码字间的最小距离必须是多少？
15. 对于下列情况，重复练习14：

    (a)$n=5$　和　$\lambda\geqslant 0.50$　　(b)$n=3$　和　$\lambda\geqslant 0.90$
16. 假设在一个二进制三重复代码中，k的值是5且集合$A=A^5$.

    (a)能够检测出多少个错误？　　(b)能够改正多少个错误？

    (c)假设$k=6$，那么(a)和(b)的答案将如何改变？
17. 练习17～22研究q进制代码(练习7). 确定类似于定理10.3的q进制代码字数量的上界.
18. 假设使用0，1，2的符号串构建一个代码，而且S是有下面性质的代码字集合：

    (i)S中的每一个字的长度都为n.

    (ii)S中的每一个字在第一个位置与最后一个位置上是2而其他位置不是.

    (iii)对于S中的每一对字a和b，有$d(a, b)=d$，其中$d(a, b)$是汉明距离.

    设T是从S通过把0变成1，而把1变成0，且保留2不变而定义的. 例如，字2102变成字2012. 579

    (a)T中两个字之间的距离是多少？

    (b)S中的字与T中的字之间的距离是多少？

    (c)假设R是由S的字加上T的字组成的. R可以检测多少个错误？可以校正多少个错误？
19. 设A是符号集$\{0, 1, 2, \cdots, q-1\}$上的$q\times q$拉丁方. 考虑由长度为3形如i, j, a_{ij}的所有串组成的q进制代码C.

    (a)如果用0取代5，确定对应于表1.4的拉丁方C.

    (b)给定任意一个$q\times q$拉丁方的代码C，这个代码可以检测出多少个错误？

    (c)它可以校正多少个错误？
20. 假设我们有秩为q的拉丁方正交族，$A^{(1)}, A^{(2)}, \cdots, A^{(r)}$，每一个拉丁方都使用符号0，1，…，$q-1$. 考虑由下面形式长度为$r+2$的所有串组成的$q$进制代码$S$：

$$i, j, a_{ij}^{(1)}, a_{ij}^{(2)}, \cdots, a_{ij}^{(r)}.$$

(a)确定对应于表 9.23 的正交族的 S.

(b)对于任意的 q，代码 S 能够检测的错误是多少？

(c)它能够校正的错误又是多少？

21. 考虑长度为 4 的代码字，这个代码字的每一个数字取自字符表{0，1，2，3，4，5}. 证明不存在 36 个代码字的集合，其中每两个代码字间的汉明距离至少是 3.

22. 设 A 是符号集合{0，1，…，$q-1$}上秩为 q 的拉丁方，而且在 9.3 节练习 5 的意义下它是水平完全的. 考虑由下面形式的长度为 4 的所有串组成的 q 元代码 T：

$$i, j, a_{ij}, a_{ij+1},$$

其中，$i=0$，1，…，$q-1$，$j=0$，1，…，$q-2$.

(a)对于 $n=4$，确定对应于 9.3 节练习 5 构造的拉丁方的 T.

(b)对于任意的 q，代码 T 可以检测出多少个错误？

(c)这个代码又能校正多少个错误？

23. 假设我们有一条 10 个数字的信息，且出现一个错误的概率 $p=0.1$.

(a)如果我们不使用代码，在传输中没有错误出现的概率是多少？

(b)如果我们使用 $k \to 3k$ 三重复代码且 $k=1$，那么正确地传输这条信息的概率是多少？

580　(c)如果我们使用大小为 2 的区组的奇偶校验码，那么正确地传输这条信息的概率是多少？

24. 假设我们有 1000 个数字的一条信息，且错误概率 $p=0.01$. 假设我们使用 $k \to 5k$ 的 5 重复代码且 $k=1$. 正确地传输这条信息的概率是多少？[提示：可以校正的错误是多少？]

25. 假设我们把多数规则解码用于 $k \to 3k$ 重复代码，且 $k=5$.

(a)如果我们接收到 auvwbcuvzbcuvwd，那么“校正”的信息是什么？

(b)如果一条信息中的一个给定数字被改变的概率是 $p=0.1$，且这一概率与这个数字在这条信息中的位置无关，那么正确地接收到长度为 $3k$ 的信息的概率是多少？

(c)继续(b)，如果我们使用多数规则解码，那么正确地解码长度 $k=5$ 的信息的概率是多少？

(d)继续(b)，如果我们使用多数规则解码，那么为使正确地解码长度 $k=5$ 的信息的概率至少达到 0.9999，我们需要的重复数量是多少？

26. (n, d)代码 C 称为**完美 t 错误校正码**(perfect t-error-correcting code)，如果 $t=\lceil (d/2)-1 \rceil$ 且不等式(10.5)成为等式. 证明这样的 C 校正的错误不会超过 t.

27. 证明 $1 \to 2t+1$ 重复代码(例子 10.5)是完美 t 错误校正码.

28. 可以证明(Tietäväinen[1973]、van Lint[1971])只存在一种除重复代码和(10.4.3 节介绍的)汉明码以外的完美 t 错误校正码. 根据 Golay[1949]，$12 \to 23$ 是一个完美 3 错误校正码. 这一代码有多少个代码字？

29. 假设 C 是最小距离为 d 的 q 进制区组 n 代码. 扩展汉明界(定理 10.3)来确定 $|C|$ 的上界.

30. 设 $\Sigma=\{0, 1\}$，$L=6$，$M=2$，且 $\chi=\{110100, 010101\}$.

(a)对于 Σ^M 中的所有可能的字 w，计算 $s_\chi(w)$.

(b)对于 $\lambda_d=(M-d)/M$，确定 Waterman 合意.

(c)对于 Σ^M 中的所有 w，计算 $\sigma_\chi(w)$.

(d)对于 Σ^M 中的所有 w，计算 $\tau_\chi(w)$.

(e)确定所有中值.

(f)确定所有平均值.

31. 对于 $\Sigma=\{0, 1\}$，$L=6$，$M=3$ 及 $X=\{000111, 011001, 100101\}$，重复练习 30.

32. (Mirkin 和 Roberts[1993])证明如果 $\lambda_d=Ad+B$，$A<0$，那么 Waterman 合意与中值过程的选择相同.

33. (Mirkin 和 Roberts[1993])证明如果 $\lambda_d = Ad^2 + B$，$A<0$，那么 Waterman 合意与平均过程的选择相同.

34. (Mirkin 和 Roberts[1993])

(a)假设 $M \geq 2$，$|\Sigma| \geq 2$，$\lambda_1 < \lambda_0$，$\lambda_d = Ad + B$，且 $A<0$. 证明对于 Σ^M 中的所有字 w 和 w' 以及 $\Sigma^L(L \geq M)$ 中不同字的所有有限集合 χ，都有 $\sigma_\chi(w) \geq \sigma_\chi(w') \leftrightarrow s_\chi(w) \leq s_\chi(w')$.

581

(b)解释这一结论.

35. (Mirkin 和 Roberts[1993])

(a)假设 $M \geq 2$，$|\Sigma| \geq 2$，$\lambda_1 < \lambda_0$，且 $\lambda_d = Ad^2 + B$，其中 $A<0$. 证明对于 Σ^M 中的所有字 w 和 w' 和 $\Sigma^L(L \geq M)$ 中不同字的所有有限集合 $\mathcal{X}$，都有 $\tau_\chi(w) \geq \tau_\chi(w') \leftrightarrow s_\chi(w) \leq s_\chi(w')$.

(b)解释这一结论.

10.4 线性代码[⊖]

10.4.1 生成矩阵

在本节，我们将看到如何使用编码的步骤去构建错误检测和错误校正码. 另外，我们给出的编码有非常高效的编码和解码过程. 我们所描述的方法很大程度上受到 R. W. Hamming 和 D. Slepian 在 20 世纪 50 年代的工作的影响. 例如，可以参见 Hamming[1950]和 Slepian[1956a，b，1960]的论文.

首先我们扩展例 10.6. 我们可以把 $k \to n$ 代码 E 考虑成是将 $n-k$ 个奇偶校验数字(parity check digit)加到长度为 k 的区组中. 一般地，信息区组(message block)$a_1 a_2 \cdots a_k$ 被编码为 $x_1 x_2 \cdots$，x_n，其中 $x_1 = a_1$，$x_2 = a_2$，…，$x_k = a_k$ 而且奇偶校验数字是由 k 个信息数字(message digit)a_1，a_2，…，a_k 决定的. 得到这样编码的最容易的方法是扩展例 10.2 的矩阵编码方法. 我们令 $\boldsymbol{M}$ 是 $k \times n$ 矩阵，并称其为生成矩阵(generator matrix)，且定义 $E(a)$ 为 $\boldsymbol{aM}$.

【例 10.7】 假设 $k=3$，$n=6$，

$$\boldsymbol{M} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

此时，信息字 110 被编码为 $110\boldsymbol{M} = 110101$. 第五个数字是 0，因为 $1+1=0$：回想一下，我们使用的是模 2 加法. 注意，$\boldsymbol{M}$ 开始于 3×3 单位矩阵. 因为我们想要对于 $i=1, 2, \cdots, k$，有 $a_i = x_i$，每一个生成矩阵都将开始于 $k \times k$ 单位矩阵 $\boldsymbol{I}_k$. ■

作为另一些例子，例 10.6 的奇偶校验码是通过取 $\boldsymbol{M}$ 为下面的矩阵来定义的：

$$\boldsymbol{M} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 1 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 1 & \cdots & 0 & 1 \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & 1 & 1 \end{bmatrix}. \tag{10.6}$$

582

我们把全都是 1 的一列加入 $k \times k$ 单位矩阵 $\boldsymbol{I}_k$. 例 10.5 的三重复代码是通过取矩阵 $\boldsymbol{M}$ 为 $k \times k$ 单位矩阵的三个拷贝而得到的.

⊖ 如果读者想要直接跳到 10.5 节，本节可以跳过.

如果一个代码是通过生成矩阵定义的，那么(传输正确时的)解码是直截了当的. 我们简单地解码 $x_1x_2\cdots x_n$ 为 $x_1x_2\cdots x_k$；即放弃 $n-k$ 个奇偶校验数字.

一般地，通过这样的生成矩阵定义的代码称为*线性代码*(linear code). 这是因为如果 $\boldsymbol{x}=x_1x_2\cdots x_n$ 和 $\boldsymbol{y}=y_1y_2\cdots y_n$ 是代码字，那么第 i 个数字是 x_i+y_i 的字 $\boldsymbol{x}+\boldsymbol{y}$ 也是代码字，其中加法是模 2 的. 为弄清楚，可简单地观察，如果 $\boldsymbol{aM}=x$，$\boldsymbol{bM}=y$，那么 $(\boldsymbol{a}+\boldsymbol{b})\boldsymbol{M}=\boldsymbol{x}+\boldsymbol{y}$.[⊖]

所以 $\boldsymbol{0}=00\cdots0$ 是每一个(非空)线性代码的代码字. 因为，如果 x 是任意一个代码字，那么由于加法是模 2 的，所以有 $\boldsymbol{x}+\boldsymbol{x}=\boldsymbol{0}$. 实际上，使用 8.2 节(请看练习 26)的术语，可以证明所有代码字的集合在运算十下定义一个群. 因此，线性代码有时候称为*二进制群代码*(binary group code).

接下来，我们注意到在线性代码中很容易确定两个代码字间的最小距离. 设位串 $\boldsymbol{x}$ 的*汉明权*(Hamming weight) $wt(\boldsymbol{x})$ 是 $\boldsymbol{x}$ 的非零数字的数量.

定理 10.5 *在线性代码 C 中，两个代码字之间的最小距离等于非零代码字的最小汉明权 ω.*

证明 注意，如果 $d(\boldsymbol{x},\boldsymbol{y})$ 是汉明距离，那么因为 C 是线性的，所以 $\boldsymbol{x}+\boldsymbol{y}$ 仍在 C 中，于是

$$d(\boldsymbol{x},\boldsymbol{y}) = wt(\boldsymbol{x}+\boldsymbol{y}).$$

这一结论使用了加法模 2 的事实. 假设对于 C 中的特定 $\boldsymbol{x}$，$\boldsymbol{y}$，最小汉明距离 d 是由 $d(\boldsymbol{x},\boldsymbol{y})$ 给出的. 那么

$$d = d(\boldsymbol{x},\boldsymbol{y}) = wt(\boldsymbol{x}+\boldsymbol{y}) \geqslant w.$$

其次，假设 C 中 $\boldsymbol{u}\neq\boldsymbol{0}$ 有最小权 w. 那么因为 $\boldsymbol{0}=\boldsymbol{u}+\boldsymbol{u}$ 仍在 C 中，我们有

$$w = wt(\boldsymbol{u}) = wt(\boldsymbol{u}+\boldsymbol{0}) = d(\boldsymbol{u},\boldsymbol{0}) \geqslant d.$$ ■

根据定理 10.5 得出在例 10.6 的代码中，两个代码字之间的最小距离等于 2，因为非 **0** 代码字的最小权等于 2. 例如，在串 11000 中可以达到这个权. 例如，这个最小距离如下：

583

$$d(11000,00000).$$

10.4.2 使用线性代码的错误校正

如果 $\boldsymbol{M}$ 是一个线性代码的生成矩阵，那么它可以示意地表示成 $[\boldsymbol{I}_k\boldsymbol{G}]$，其中 $\boldsymbol{G}$ 是一个 $k\times(n-k)$ 矩阵. 设 $\boldsymbol{G}^{\mathrm{T}}$ 是矩阵 $\boldsymbol{G}$ 的转置矩阵且设 $\boldsymbol{H}=[\boldsymbol{G}^{\mathrm{T}}\boldsymbol{I}_{n-k}]$ 是一个 $(n-k)\times n$ 矩阵，称为*奇偶校验矩阵*(parity check matrix). 在例 10.6 中，矩阵 $\boldsymbol{G}^{\mathrm{T}}$ 是 k 个 1 的行向量，而矩阵 $\boldsymbol{H}$ 是 $(k+1)$ 个 1 的行向量. 在例 10.5 中，使用三次重复及长度为 4 的信息区组，我们有

$$\boldsymbol{G}^{\mathrm{T}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

且

⊖ 在本节中，我们假设 $A=A^k$.

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{10.7}$$

在例 10.7 中，

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}. \tag{10.8}$$

注意，我们可以通过由生成矩阵 M 给出的矩阵 G 或奇偶校验矩阵 H 来定义线性代码. 因为，我们可以从 G 得到 H 及从 H 得到 G. 我们将看到奇偶校验矩阵将提供一种检测及校正错误的方法. 这一方法的基本思想归功于 Slepian[1956a, b, 1960].

首先，我们注意到，奇偶校验矩阵可以用于鉴别代码字.

定理 10.6 *在线性代码中，区组 $\boldsymbol{a}=a_1a_2\cdots a_k$ 被编码成 $\boldsymbol{x}=x_1x_2\cdots x_n$，当且仅当对于 $i\leqslant k$，$a_i=x_i$ 且 $\boldsymbol{Hx}^{\mathrm{T}}=\boldsymbol{0}$.*

证明[⊖] 根据编码的定义，如果区组 $\boldsymbol{a}=a_1a_2\cdots a_k$ 被编码成 $\boldsymbol{x}=x_1x_2\cdots x_n$，那么

$$(a_1a_2\cdots a_k)[\boldsymbol{I}_k\boldsymbol{G}]=(x_1x_2\cdots x_n).$$

584

现在，显然对于 $i\leqslant k$，有 $a_i=x_i$. 而且，

$$\boldsymbol{Hx}^{\mathrm{T}}=\boldsymbol{H}(\boldsymbol{a}[\boldsymbol{I}_k\boldsymbol{G}])^{\mathrm{T}}=\boldsymbol{H}[\boldsymbol{I}_k\boldsymbol{G}]^{\mathrm{T}}\boldsymbol{a}^{\mathrm{T}}=[\boldsymbol{G}^{\mathrm{T}}\boldsymbol{I}_{n-k}]\begin{bmatrix}\boldsymbol{I}_k\\ \boldsymbol{G}^{\mathrm{T}}\end{bmatrix}\boldsymbol{a}^{\mathrm{T}}$$
$$=(\boldsymbol{G}^{\mathrm{T}}+\boldsymbol{G}^{\mathrm{T}})\boldsymbol{a}^{\mathrm{T}}=\boldsymbol{0}\boldsymbol{a}^{\mathrm{T}}=\boldsymbol{0},$$

其中 $\boldsymbol{G}^{\mathrm{T}}+\boldsymbol{G}^{\mathrm{T}}=\boldsymbol{0}$，因为加法是模 2 的.

反过来，假设对于 $i\leqslant k$ 有 $a_i=x_i$，且 $\boldsymbol{Hx}^{\mathrm{T}}=\boldsymbol{0}$. 现在，假设 a 被编码为 $\boldsymbol{y}=y_1y_2\cdots y_n$. 那么，对于 $i\leqslant k$，$a_i=y_i$. 所以，对于 $i\leqslant k$，$x_i=y_i$. 同样，根据证明的第一部分，$\boldsymbol{Hy}^{\mathrm{T}}=\boldsymbol{0}$. 所以有 $x=y$ 成立. 为了说明其原因，注意，等式 $\boldsymbol{Hx}^{\mathrm{T}}=\boldsymbol{0}$ 和 $\boldsymbol{Hy}^{\mathrm{T}}=\boldsymbol{0}$ 中的每一个都给出一个方程组. 对于第一种情况，第 j 个方程至多包含变量 x_1, x_2, …, x_k 和 x_{k+j}. 因此，因为 x_1, x_2, …, x_k 是由 a_1, a_2, …, a_k 给出的，所以第 j 个方程使用 k 个信息数字 a_1, a_2, …, a_k 唯一地定义 x_{k+j}. 这一论述对 y_{k+j} 同样适用. 因此，$x_{k+j}=y_{k+j}$. ■

推论 10.6.1 *位串 $\boldsymbol{x}=x_1x_2\cdots x_n$ 是代码字当且仅当 $\boldsymbol{Hx}^{\mathrm{T}}=\boldsymbol{0}$.*

推论 10.6.2 *存在唯一位串 $\boldsymbol{x}$ 使得对于 $i\leqslant k$ 有 $x_i=a_i$，且 $\boldsymbol{Hx}^{\mathrm{T}}=\boldsymbol{0}$. 这个位串 x 是 $a_1a_2\cdots a_k$ 的编码. 使用 a_1, a_2, …, a_k 表示 $\boldsymbol{x}$ 的项 x_{k+j}，其表达式是把 $\boldsymbol{H}$ 的第 j 行乘以 $\boldsymbol{x}^{\mathrm{T}}$.*

证明 这一结论是定理证明的推论.

为了说明这些结果，注意，在奇偶校验代码中，$H=(11\cdots1)$ 有 $n=k+1$ 个 1. 这时，$\boldsymbol{Hx}^{\mathrm{T}}=\boldsymbol{0}$ 当且仅当

$$x_1+x_2+\cdots+x_{k+1}=0. \tag{10.9}$$

但是，因为加法是模 2 的，所以(10.9)式正好等价于下面的条件：

⊖ 本证明可以跳过.

$$x_{k+1} = x_1 + x_2 + \cdots + x_k,$$

这个方程定义代码字，称为奇偶校验方程(parity check equation). 类似地，在 $k=4$ 的三重复代码中，我们从(10.7)式看到 $\boldsymbol{H}\boldsymbol{x}^{\mathrm{T}}=\boldsymbol{0}$ 当且仅当

$$\begin{aligned} &x_1 + x_5 = 0, \quad x_2 + x_6 = 0, \quad x_3 + x_7 = 0, \quad x_4 + x_8 = 0, \\ &x_1 + x_9 = 0, \quad x_2 + x_{10} = 0, \quad x_3 + x_{11} = 0, \quad x_4 + x_{12} = 0. \end{aligned} \tag{10.10}$$

因为加法是模 2 的，所以方程(10.10)等价于

$$\begin{aligned} &x_1 = x_5, \quad x_2 = x_6, \quad x_3 = x_7, \quad x_4 = x_8, \\ &x_1 = x_9, \quad x_2 = x_{10}, \quad x_3 = x_{11}, \quad x_4 = x_{12}, \end{aligned}$$

585

上述方程定义代码字，称为奇偶校验方程. 最后，在例 10.7 中，通过(10.8)式，$\boldsymbol{H}\boldsymbol{x}^{\mathrm{T}}=\boldsymbol{0}$ 当且仅当

$$\begin{aligned} x_1 + x_3 + x_4 &= 0, \\ x_1 + x_2 + x_5 &= 0, \\ x_2 + x_3 + x_6 &= 0. \end{aligned}$$

利用加法是模 2 的事实，这些方程等价于下面的奇偶校验方程，这些方程使用下面的信息数字 $x_i(i \leqslant k)$ 确定奇偶校验数字 x_{k+j}：

$$\begin{aligned} x_4 &= x_1 + x_3, \\ x_5 &= x_1 + x_2, \\ x_6 &= x_2 + x_3. \end{aligned}$$

因此，正如我们前面所看到的那样，110 被编码为 110101，因为 $x_4=1+0=1$，$x_5=1+1=0$，$x_6=1+0=1$. ■

一般地，根据推论 10.6.2，方程 $\boldsymbol{H}\boldsymbol{x}^{\mathrm{T}}=\boldsymbol{0}$ 使用 $a_1, a_2, \cdots, a_k$ 定义 x_{k+j}. 利用这些项给出 x_{k+j} 的方程是奇偶校验方程.

定理 10.6 和推论 10.6.1 给出一种检测错误的方法. 我们注意到，如果接收到 $\boldsymbol{x}$ 且 $\boldsymbol{H}\boldsymbol{x}^{\mathrm{T}} \neq \boldsymbol{0}$，那么就出现了错误. 奇偶校验矩阵使得我们不仅可以检测出错误，而且可以校正这些错误，正如下面的定理所给出的那样.

定理 10.7　假设奇偶校验矩阵 $\boldsymbol{H}$ 的各列都是非零的且各不相同. 假设传输了代码字 $\boldsymbol{y}$ 且接收到字 $\boldsymbol{x}$. 如果 $\boldsymbol{x}$ 只在第 i 个数字上不同于 $\boldsymbol{y}$，那么 $\boldsymbol{H}\boldsymbol{x}^{\mathrm{T}}$ 是 $\boldsymbol{H}$ 的第 i 列.

证明　注意，根据推论 10.6.1，$\boldsymbol{H}\boldsymbol{y}^{\mathrm{T}}=\boldsymbol{0}$. 现在 $\boldsymbol{x}$ 可以写成 $\boldsymbol{y}+\boldsymbol{e}$. 其中 $\boldsymbol{e}$ 是错误串，即在不同于 $\boldsymbol{y}$ 的位上是 1 的位串.（回想一下加法是模 2 的.）我们得出结论：

$$\boldsymbol{H}\boldsymbol{x}^{\mathrm{T}} = \boldsymbol{H}(\boldsymbol{y}+\boldsymbol{e})^{\mathrm{T}} = \boldsymbol{H}(\boldsymbol{y}^{\mathrm{T}}+\boldsymbol{e}^{\mathrm{T}}) = \boldsymbol{H}\boldsymbol{y}^{\mathrm{T}} + \boldsymbol{H}\boldsymbol{e}^{\mathrm{T}} = \boldsymbol{H}\boldsymbol{e}^{\mathrm{T}}.$$

现在，如果 $\boldsymbol{e}$ 是除第 i 个位置上是 1 之外所有位上都是 0 的向量，那么 $\boldsymbol{H}\boldsymbol{e}^{\mathrm{T}}$ 是 $\boldsymbol{H}$ 的第 i 列. ■

为了说明这一结果，假设我们有一个 $k=4$ 的三重复代码. 那么奇偶校验矩阵 $\boldsymbol{H}$ 由(10.7)式给出. 假设发送代码字 $\boldsymbol{y}=110111011101$，而接收的字 $\boldsymbol{x}=110011011101$，它在第 4 个位置上不同于 y. 那么注意

586

$$\boldsymbol{H}\boldsymbol{x}^{\mathrm{T}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

这是 $\boldsymbol{H}$ 的第 4 列.

一般地，使用奇偶校验矩阵 $\boldsymbol{H}$ 的错误校正将按如下的过程进行：假设 $\boldsymbol{H}$ 的各列不同且都是非 0 的. 接收到的区组是 $\boldsymbol{x}$ 时，计算 $\boldsymbol{H}\boldsymbol{x}^{\mathrm{T}}$. 如果计算结果是 $\mathbf{0}$，那么传输中出现错误的可能性不大，因此可以合理地假设 $\boldsymbol{x}$ 是正确的. 如果其计算结果不是 $\mathbf{0}$，而是 $\boldsymbol{H}$ 的第 i 列，而且如果出现多个错误的可能性不大，那么可以合理地假设出现一个错误，所以正确的字在第 i 个位上不同于 $\boldsymbol{x}$. 如果 $\boldsymbol{H}\boldsymbol{x}^{\mathrm{T}}$ 不等于 0 且不是 $\boldsymbol{H}$ 的一个列，那么在传输中至少出现了两个错误，而且不能使用这种方法进行错误校正.

10.4.3 汉明码

回想一下，对于错误校正，我们希望 $(n-k)\times n$ 奇偶校验矩阵 $\boldsymbol{H}$ 有非 0 的且各不相同的列. 现在，如果 $p=n-k$，$\boldsymbol{H}$ 的列是长度为 p 的位串. 存在 2^p 个这样的串，2^p-1 个非 0 串. 如果我们取 H 为这样的矩阵：它的列是所有 2^p-1 个非 0 的长度为 p 的位串，且按照任意顺序排列，这时我们得到汉明码(Hamming code) $\mathcal{H}_p$. 从技术上讲，为了与我们的定义一致，后 $n-k$ 列应该形成单位矩阵. 上面的结果代码是 $k\to n$ 代码，其中 $n=2^p-1$，$k=n-p=2^p-1-p$. 例如，如果 $p=2$，那么 $n=2^2-1=3$ 且典型的 $\boldsymbol{H}$ 可以是由下面的矩阵给出的：

$$\boldsymbol{H}=\begin{bmatrix}1 & 1 & 0\\ 1 & 0 & 1\end{bmatrix}.$$

很容易看出 $\boldsymbol{H}$ 定义 $1\to3$ 三重复代码. 如果 $p=3$，那么 $n=2^3-1=7$，且典型的 $\boldsymbol{H}$ 可以由下面的矩阵给出：

$$\boldsymbol{H}=\begin{bmatrix}0 & 1 & 1 & 1 & 1 & 0 & 0\\ 1 & 0 & 1 & 1 & 0 & 1 & 0\\ 1 & 1 & 0 & 1 & 0 & 0 & 1\end{bmatrix}.$$

我们得到一个 $4\to7$ 代码. Hamming[1950]和 Golay[1949]引入了汉明码. [注意如果改变 $\boldsymbol{H}$ 中列的顺序，我们得到(实质上)相同的代码，或代码字集合. 因为通过改变奇偶校验数字的顺序，任意两个这样的代码(在练习 25 的定义的意义下)被看成是等价的. 这就是都称其为汉明码 $\mathcal{H}_p$ 的原因.]

587

定理 10.8 *对于 $p\geqslant2$，在汉明码 $\mathcal{H}_p$ 中，两个代码字之间的最小距离是 3.*

证明 因为奇偶校验矩阵 $\boldsymbol{H}$ 的列是非 0 且互不相同的，所以可以校正单一的错误(参见定理 10.7 后面的讨论). 因此，根据定理 10.2，$d\geqslant3$. 现在，容易证明，对于 $p\geqslant2$，总是存在 3 个长度为 p 的非 0 位串，(在模 2 加法下)它们的和等于 0(练习 23). 如果这三个位串作为矩阵 $\boldsymbol{H}$ 的列 u，v 和 w 出现，我们取 x 是这样的向量：在 u，v，w 的位置上是 1，而其他位置是 0. 那么 $\boldsymbol{H}\boldsymbol{x}^{\mathrm{T}}$ 是 $\boldsymbol{H}$ 的第 u 列，第 v 列和第 w 列的和，所以等于 0. 因此，$\boldsymbol{x}$ 是权为 3 的代码字，根据定理 10.5，$d\leqslant3$. ■

所以，根据定理 10.8 可知，汉明码总可以检测出至多两个错误并可以校正至多一个错误. 在练习 27 中，我们要求读者证明汉明码校正的错误数量不可能大于 1.

本节练习

1. 假设有下面的矩阵 $\boldsymbol{M}$：

$$\boldsymbol{M}=\begin{bmatrix}1 & 0 & 1 & 1\\ 0 & 1 & 1 & 0\end{bmatrix}.$$

使用 $\boldsymbol{M}$，确定对应于下面每一个信息字 $a_1a_2\cdots a_k$ 的代码字 $x_1x_2\cdots x_n$.

(a) 11　　(b) 10　　(c) 01　　(d) 00

2. 假设有下面矩阵 $\boldsymbol{M}$:

$$\boldsymbol{M}=\begin{bmatrix}1&0&0&1&0&0\\0&1&0&0&1&0\\0&0&1&0&0&1\end{bmatrix}.$$

对于下面的信息字重复练习 1.

(a)111　　(b)101　　(c)000

3. 假设有下面的矩阵 $\boldsymbol{M}$:

$$\boldsymbol{M}=\begin{bmatrix}1&0&0&0&0&1&1\\0&1&0&0&1&0&1\\0&0&1&0&1&1&0\\0&0&0&1&1&1&1\end{bmatrix}.$$

对于下面的信息字重复练习 1.

(a)1111　　(b)1000　　(c)0001

4. 确定由下面矩阵 $\boldsymbol{M}$ 所生成的线性代码 C:

(a)练习 1　　(b)练习 2　　(c)练习 3

588

5. 确定把 0 翻译成 000000 而把 1 翻译成 111111 的代码的生成矩阵.

6. 下面的每一代码都是线性代码. 对于每一个代码，确定两个代码字之间的最小距离.

(a)000000，001001，010010，100100，011011，101101，110110，111111

(b)0000，0011，0101，1001，0110，1010，1100，1111

(c)00000,00011，00101，01001，10001，00110，01010，01100，10010，10100，11000，01111，11011，10111，11101，11110

(d)11111111,10101010，11001100，10011001，11110000，10100101，11000011，10010110，00000000，01010101，00110011，01100110，00001111，01011010，00111100，01101001

7. 确定对应于下面的矩阵 $\boldsymbol{M}$ 的奇偶校验矩阵 $\boldsymbol{H}$:

(a)练习 1　　(b)练习 2　　(c)练习 3

8. 确定对应于下面的奇偶校验矩阵 $\boldsymbol{H}$ 的生成矩阵 $\boldsymbol{M}$:

(a)$\boldsymbol{H}=\begin{bmatrix}1&1&0&1&0&0\\1&0&1&0&1&0\\0&1&1&0&0&1\end{bmatrix}$　　(b)$\boldsymbol{H}=\begin{bmatrix}1&1&1&0&0\\1&0&0&1&0\\1&1&0&0&1\end{bmatrix}$　　(c)$\boldsymbol{H}=\begin{bmatrix}1&0&1&0&0&0\\1&1&0&1&0&0\\0&1&0&0&1&0\\0&0&0&0&0&1\end{bmatrix}$.

9. 确定下列每一个练习中的奇偶校验方程:

(a)练习 1　　(b)练习 2　　(c)练习 3

(d)练习 8(a)　　(e)练习 8(b)　　(f)练习 8(c)

10. 假设我们使用 $k\rightarrow 5k$ 的 5 重复代码，其中 $k=2$. 确定奇偶校验矩阵 $\boldsymbol{H}$，推导奇偶校验方程.

11. 对于练习 8(a)中的矩阵 $\boldsymbol{H}$，假设通过一个吵杂的信道接收到字 $\boldsymbol{x}$. 假设发生错误的可能性不大. 对于下面的每一个 $\boldsymbol{x}$，确定在传输过程中是否产生了错误.

(a)$\boldsymbol{x}$=111000　　(b)$\boldsymbol{x}$=111100　　(c)$\boldsymbol{x}$=000100

12. 对于练习 8(b)的 $\boldsymbol{H}$ 和下面的 $\boldsymbol{x}$ 重复练习 11.

(a)$\boldsymbol{x}$=11101　　(b)$\boldsymbol{x}$=01101　　(c)$\boldsymbol{x}$=00011

13. 对于练习 1 的 $\boldsymbol{M}$ 和下面的 $\boldsymbol{x}$ 重复练习 11.

(a)$\boldsymbol{x}$=1111　　(b)$\boldsymbol{x}$=1000　　(c)$\boldsymbol{x}$=0100

14. 对于练习 11 的每一个部分，如果有可能，在存在一个错误的情况下校正它.

589

15. 对于练习 12 的每一个部分，如果有可能，在存在一个错误的情况下校正它.

16. 考虑奇偶校验矩阵

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

确定对应于代码中的所有代码字. 这个代码校正所有单一错误吗?

17. 在 $p=3$ 的汉明码$\mathcal{H}_p$ 中，编码下面的字.
(a)1001　　(b)0001　　(c)1110
18. 对于 $p=4$ 的汉明码$\mathcal{H}_p$，确定奇偶校验矩阵并编码字 10000000000.
19. 确定汉明码$\mathcal{H}_2$的所有代码字.
20. 确定汉明码$\mathcal{H}_3$ 的所有代码字.
21. 对于每一个 $p\geqslant 2$，在汉明码$\mathcal{H}_p$ 中，确定一个权为 3 的字.
22. 对于 $p\geqslant 2$，在汉明码$\mathcal{H}_p$ 中考虑代码字 $x_1x_2\cdots x_n$. 设 $k=n-p$.
(a)证明对于所有 $i\leqslant k$，如果 $x_i=0$，那么对于所有的 i，$x_i=0$.
(b)证明不存在正好有一个 x_i 等于 1 而所有其他的 x_i 等于 0 的情况.
23. 证明如果 $p\geqslant 2$，那么总存在长度为 p 的 3 个位串，它们的和(在模 2 加法下)等于 0.
24. 设 C 是一个线性代码，其代码字的长度是 n，而且其中有一些代码字有奇数权. 通过把 0 加入到有偶数权的 C 的每一个字的末端，而把 1 加入到有奇数权的 C 的每一个字的末端，形成一个新代码 C'.
(a)如果 M 是 C 的奇偶校验矩阵，证明 C'的奇偶校验矩阵由下面的矩阵给出：

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ & & & 0 \\ & M & & 0 \\ & & & \vdots \\ & & & 0 \end{bmatrix}.$$

(b)证明 C'的任意两个代码字之间的距离是偶数.
(c)证明：如果 C 的两个代码字之间的最小距离 d 是奇数，那么 C'的两个代码字之间的最小距离是$d+1$.
(d)对于所有 $p\geqslant 2$，确定一个线性代码，其代码字的长度是 $n=2^p$ 且两个代码字之间的最小距离等于 4.
25. 两个 n 代码 C 和 C'是等价的(equivalent)，如果它们仅在符号的顺序上不同，即如果一个代码可以从另一个代码，通过对每一个元素运用相同的排列而得到的话. 例如，下面两个代码 C 和 C'等价：

C	C'
0000	0000
1111	1111
0011	0101
1100	1010

590

证明：如果 M 是生成代码 C 的生成矩阵，而且 M'是从 M 通过互换对应于 M 的矩阵 G 的行得到的，而且 M'生成代码 C'，那么 C 和 C'是等价的.
26. 证明在线性代码下，在运算+下代码字的集合定义一个群.
27. 证明对于 $p\geqslant 2$，汉明码$\mathcal{H}_p$ 是完美 1 错误校正码(参见 10.3 节练习 26 中完美 t 错误校正码的定义).
28. 我们可以把线性代码理论从二进制代码扩展到 q 进制代码上(参见 10.3 节的练习 7、练习 13 和练习 17～22)，其中 q 是素数的幂. 我们继续从信息字通过加入奇偶校验数字来定义代码字，而且特别地使用生成矩阵 M，但是使用有限域 GF(q)中的加法，而不是模 2 加法. 注意，在练习 29 和练习 30 中说明了主要例外，我们的大部分结果对一般的情况都成立. 假设 $q=5$.
(a)如果 M 如练习 1 中的矩阵，确定对应于下面信息字的代码字：
i. 14　　ii. 03　　iii. 13
(b)如果 M 是练习 2 中的矩阵，确定对应于下面信息字的代码字：
i. 124　　ii. 102　　iii. 432

29. 继续练习 28.

 (a)注意 $d(\boldsymbol{x},\ \boldsymbol{y})$不一定是 $wt(\boldsymbol{x}+\boldsymbol{y})$.

 (b)距离和权之间的关系是什么?

30. 继续练习 28，证明定理 10.6 不一定为真.

31. 继续练习 28，证明定理 10.5 仍然成立.

32. 假设 C 是最小距离为 d 的 q 进制区组 n 代码. 扩展汉明界(定理 10.3)来确定 $|C|$ 的上界.

10.5 运用区组设计确定错误校正码[⊖]

10.5.1 阿达马码

寻找错误校正码的一个方法是，首先集中精力寻找一个丰富的代码字集合 C，然后在 C 中实施信息编码. 这与我们迄今给出的方法恰好相反，前面给出的方法是首先定义编码，然后把这一编码所产生的所有字的集合定义为集合 C. 我们的目标是寻找这样的代码字集合 C：校正给定数量的错误，或等价地，两个代码字之间有给定最小距离 d，且在该集合中有很多代码字，因此，允许更多可能的信息区组被编码.

回想一下，一个$(n,\ d)$代码有长度为 n 的代码字，而且两个代码字之间的最小距离是 d. 构建$(n,\ d)$代码的一个有效方法是使用$(v,\ k,\ \lambda)$设计的关联矩阵(参见 9.4.2 节). 这个关联矩阵的每一行有 $r=k$ 个 1，而其余项都是 0. 行的长度为 $b=v$. 任意两行在一列上同时为 1 的数量正好是 λ 个. 对于 $n=v$，这样的行可以定义$(n,\ d)$代码的代码字. 那么，d 是多少? 让我们考虑两行，比如说第 i 行和第 j 行. 存在 λ 个列，在这些列上，这两个行有 1. 在每一个行上有 k 个 1，因此存在 $k-\lambda$ 个列，在这些列上行 i 有 1 而行 j 有 0，而且存在 $k-\lambda$ 个列，其中行 i 有 0，行 j 有 1. 所有其他的列在两行上有 0. 所以这两行有 $2(k-\lambda)$个位置不同. 这一结论对每一对行都成立，所以

$$d=2(k-\lambda).$$

定理 9.17 表明，对于任意大的 m，存在维数 m 的阿达马设计，即$(4m-1,\ 2m-1,\ m-1)$设计. 所以，对于任意大的 $k-\lambda$，我们可以找到$(v,\ k,\ \lambda)$设计，因此，对于任意大的 d，有$(n,\ d)$代码. 对于给定维数 m 的阿达马设计，我们有：

$$d=2(k-\lambda)=2[(2m-1)-(m-1)]=2m.$$

因此，如果对于任意大的 m 存在维数 m 的阿达马设计，那么存在错误校正码，对于任意大的 m，这个代码检测至多 $d-1=2m-1$ 个错误并校对至多$\lceil (d/2)-1\rceil=m-1$ 个错误. 这些代码是$(4m-1,\ 2m)$代码，因为每一个代码字有长度 $4m-1$. 我们称它们为*阿达马码*(Hadamard code). 我们首先要证明任意的大维数 m 的阿达马设计的存在，即要证明定理 9.17. 然后，我们考察由阿达马设计的关联矩阵所构建的代码的丰富程度；即这些代码与最丰富的$(n,\ d)$代码相比如何?

10.5.2 构建阿达马设计[⊜]

构建阿达马设计的基本思想是，特定类型的矩阵给出这些设计的关联矩阵. 一个 $n\times n$ 矩阵 $\boldsymbol{H}=(h_{ij})$被称为*秩为 n 的阿达马矩阵*(Hadamard matrix of order n)，如果对于每一个 i 和 j 有 h_{ij} 等于$+1$ 或者等于-1，且如果

$$\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}=n\boldsymbol{I},$$

⊖ 本节依赖于 9.4 节的内容.

⊜ 本小节可以跳过.

其中 $\boldsymbol{H}^{\mathrm{T}}$ 是 $\boldsymbol{H}$ 的转置矩阵，$\boldsymbol{I}$ 是 $n\times n$ 单位矩阵⊖. 矩阵 $n\boldsymbol{I}$ 在对角线下方为 n，其余位置为 0. 作为例子，假设

$$\boldsymbol{H}=\begin{bmatrix}1 & 1\\ -1 & 1\end{bmatrix}. \tag{10.11}$$

于是有

$$\boldsymbol{H}^{\mathrm{T}}=\begin{bmatrix}1 & -1\\ 1 & 1\end{bmatrix},$$

592

且

$$\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}=\begin{bmatrix}2 & 0\\ 0 & 2\end{bmatrix}.$$

因此，(10.11)式的矩阵 $\boldsymbol{H}$ 是秩为 2 的阿达马矩阵. 一个秩为 4 的阿达马矩阵由下面的矩阵给出：

$$\boldsymbol{H}=\begin{bmatrix}1 & 1 & -1 & 1\\ 1 & -1 & -1 & -1\\ -1 & -1 & -1 & 1\\ 1 & -1 & 1 & 1\end{bmatrix}. \tag{10.12}$$

因为容易验证

$$\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}=\begin{bmatrix}4 & 0 & 0 & 0\\ 0 & 4 & 0 & 0\\ 0 & 0 & 4 & 0\\ 0 & 0 & 0 & 4\end{bmatrix}.$$

阿达马矩阵的一个等价定义如下所述. 回想一下，如果

$$(x_1,x_2,\cdots,x_n) \quad 和 \quad (y_1,y_2,\cdots,y_n)$$

是两个向量，那么它们的内积(inner product)由下式定义：

$$\sum_i x_i y_i.$$

于是，$+1$ 和 -1 构成的 $n\times n$ 矩阵是阿达马矩阵，如果对于所有的 i，第 i 行与其自身的内积是 n，且对于所有 $i\neq j$，第 i 行与第 j 行的内积等于 0. 这正是阿达马矩阵的定义的重述. 为了说明这一点，我们考虑(10.11)式的矩阵 $\boldsymbol{H}$. 这个矩阵的第一行是向量(1, 1)，而第二行是向量(−1, 1). 第一行与其自身的内积是

$$1\cdot 1+1\cdot 1=2.$$

第一行与第二行的内积是

$$1\cdot(-1)+1\cdot 1=0$$

等等.

一个阿达马矩阵称为是规范的(nomalized)，如果它的第一行和第一列只由 $+1$ 组成. 例如，矩阵

$$\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \tag{10.13}$$

是规范阿达马矩阵. 而矩阵

⊖ 阿达马矩阵是由 Hadamard[1893]提出的，而且更早时，Sylvester[1867]使用了这样的矩阵. Plotkin[1960]、Bose 和 Shrikhande[1959]由阿达马矩阵构建了二进制代码.

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{10.14}$$

593

也是规范阿达马矩阵.

阿达马矩阵的若干重要性质概括为如下定理.

定理 10.9 如果 $\boldsymbol{H}$ 是秩为 $n>2$ 的规范阿达马矩阵，那么对于某个 m 有 $n=4m$. 而且，除第一行(列)之外的每一行(列)正好有 $2m$ 个 $+1$ 和 $2m$ 个 -1. 而且对于除第一行(列)之外的任意两行(列)，都正好存在 m 列(行)，在这些列(行)上两行(列)都是 $+1$.

我们将在本小节的后面给出定理 10.9 的证明. 这里，我们来看看定理 10.9 是如何给出定理 9.17 的证明的. 给定一个规范阿达马矩阵，我们可以定义一个(v, k, λ)设计. 通过删除第一行和第一列，在其剩余部分，把每一个 -1 换成 0 就可以做到这一点. 正如我们将看到的那样，这样做给出一个(v, k, λ)设计的关联矩阵. 在(10.14)式的阿达马矩阵上进行这一过程，首先给出

$$\begin{bmatrix} -1 & 1 & -1 \\ 1 & -1 & -1 \\ -1 & -1 & 1 \end{bmatrix},$$

于是这个关联矩阵是

$$\boldsymbol{A} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} B_1 & B_2 & B_3 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array},$$

这给出下面的区组：

$$B_1 = \{2\}, \quad B_2 = \{1\}, \quad B_3 = \{3\},$$

而且产生一个 $v=3$，$k=1$，$\lambda=0$ 的设计(从技术角度看，这不是一个设计，因为我们要求 $\lambda>0$. 然而，这可以说明该过程).

为了证明这一过程总能给出(v, k, λ)设计，需要注意，根据定理 10.9，关联矩阵 $\boldsymbol{A}$ 有 $4m-1$ 个行和 $4m-1$ 个列，所以 $b=v=4m-1$. 同样，从每一行消去一个 1(第一个 1)，所以 $\boldsymbol{A}$ 的每一行有 $2m-1$ 个 1，且 $r=2m-1$. 通过类似的讨论，$\boldsymbol{A}$ 的每一列有 $2m-1$ 个 1 且 $k=2m-1$. 最后，任意两行在第一列上同时有两个 1，所以现在同时有一个更少的对，即 $m-1$ 对. 所以，$\lambda=m-1$. 因此，我们有(v, k, λ)设计，其中 $v=4m-1$，$k=2m-1$，$\lambda=m-1$，且 $m\geqslant 2$.[⊖]

事实上，我们所描述的这一过程是可逆的. 我们有下面的定理.

594

定理 10.10 存在维数 m 的阿达马设计，当且仅当存在秩为 $4m$ 的阿达马矩阵.

证明 还是从一个维数为 m 的阿达马矩阵开始，并从这个矩阵构建一个秩为 $4m$ 的阿达马矩阵就可以了. 设 $\boldsymbol{A}$ 是这样设计的关联矩阵. 通过把 0 变成 -1，并把一行 1 加入到矩阵的顶层，把一例 1 加到前面，构建矩阵 $\boldsymbol{H}$. 容易证明 $\boldsymbol{H}$ 是阿达马矩阵. 这一证明留作练习(练习 14). ■

接下来，我们来看，对于任意大的 m，如何构建秩为 $4m$ 的规范阿达马矩阵. 这将给出定理 9.17 的证明. 假设 $\boldsymbol{H}$ 是一个 $n\times n$ 阿达马矩阵. 设 $\boldsymbol{K}$ 是如下矩阵：

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{H} & \boldsymbol{H} \\ \boldsymbol{H} & -\boldsymbol{H} \end{bmatrix},$$

⊖ 如果 $m=1$，正如我们看到的那样，有 λ 等于 0.

其中，$-\boldsymbol{H}$ 是根据矩阵 $\boldsymbol{H}$ 把其各项都乘以 -1 而得的矩阵. 例如，如果 $\boldsymbol{H}$ 是(10.13)式的矩阵，那么 $\boldsymbol{K}$ 是下面的矩阵：

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

$\boldsymbol{K}$ 总是秩为 $2n$ 的阿达马矩阵，因为它的每一项是 $+1$ 或者 -1. 另外，容易证明行与其自身的内积总等于 $2n$，两个不同行的内积总等于 0. 最后，我们发现如果 $\boldsymbol{H}$ 是规范的，那么 $\boldsymbol{K}$ 也是规范的. 因此，存在任意大秩的规范阿达马矩阵，特别地，对于任意的 $p\geqslant 1$，存在所有秩为 2^p 的规范阿达马矩阵. 对于 $m\geqslant 2$，秩为 $4m$ 的阿达马矩阵给出$(4m-1,\ 2m-1,\ m-1)$设计. 因为 $4m=2^p$，$m\geqslant 2$ 等价于 $m=2^k$，其中 $k\geqslant 1$，这样就完成了定理 9.17 的证明.

如果矩阵 $\boldsymbol{H}$ 是(10.14)式所示的矩阵，继续我们刚才讨论的构建过程是很有趣的. 这时，对应的矩阵 $\boldsymbol{K}$ 是下面的矩阵：

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}. \tag{10.15}$$

所对应的$(v,\ k,\ \lambda)$设计的关联矩阵是通过删除第一行和第一列并把剩余的 -1 变成 0 而得到的. 于是我们有

595

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}. \tag{10.16}$$

这是 $v=7$，$k=3$，$\lambda=1$ 时的对称 BIBD 的关联矩阵. 它定义了不同于例 9.10 的设计. 再一次重复这一构造，我们得到一个 16×16 阿达马矩阵，因此得到一个(15，7，3)设计. 这个设计就是表 9.27 所示的区组设计. 证明作为练习 8 留给读者.

对于任意大的 m，我们概述的这一构造过程一定产生秩为 $4m$ 的规范阿达马矩阵，特别对于 $m=1，2，4，8，\cdots$，所得的秩为 4，8，16，32，$\cdots$. 注意，我们不能肯定对于每一个 m，可以构造出秩为 $4m$ 的阿达马规范矩阵. 人们猜测，对于每一个 m，存在秩为 $4m$ 的规范阿达马矩阵. 这一猜测是否为真是组合设计和编码数字的另一个未解问题. 注意，如果存在任何秩为 $4m$ 的阿达马矩阵，那么一定存在规范阿达马矩阵(练习 6)(关于构造阿达马矩阵的其他方法，可以参见 Hall[1967]或 MacWilliams 和 Solane[1983]. Craigen 和 Wallis[1993]是一篇关于阿达马矩阵的概述文章，作为这样的矩阵被提出 100 周年的纪念).

我们通过补全定理 9.17 的证明中遗漏的一个步骤来结束本小节，这一遗漏步骤就是要证明

定理 10.9[⊖]. 我们需要一个预备结果.

定理 10.11　如果 $\boldsymbol{H}$ 是阿达马矩阵，那么 $\boldsymbol{H}^{\mathrm{T}}$ 也是阿达马矩阵.

证明　如果 $\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}=n\boldsymbol{I}$，那么

$$\frac{\boldsymbol{H}}{\sqrt{n}}\frac{\boldsymbol{H}^{\mathrm{T}}}{\sqrt{n}}=\boldsymbol{I},$$

其中 $\boldsymbol{H}/\sqrt{n}$ 可以从 $\boldsymbol{H}$ 通过把它的每个项除以 $\sqrt{n}$ 得到，类似地，可以得到 $\boldsymbol{H}^{\mathrm{T}}/\sqrt{n}$. 因为对于方阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$，$\boldsymbol{AB}=\boldsymbol{I}$ 蕴涵 $\boldsymbol{BA}=\boldsymbol{I}$，所以我们有

$$\frac{\boldsymbol{H}^{\mathrm{T}}}{\sqrt{n}}\frac{\boldsymbol{H}}{\sqrt{n}}=\boldsymbol{I},$$

或

$$\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}=n\boldsymbol{I}.$$

因为 $(\boldsymbol{H}^{\mathrm{T}})^{\mathrm{T}}=\boldsymbol{H}$，所以 $\boldsymbol{H}^{\mathrm{T}}$ 是阿达马矩阵. ■

定理 10.9 的证明　设 $\boldsymbol{H}$ 是秩为 n 的规范阿达马矩阵. 通过运用定理 10.11，原来列的结果是根据行的结果得出的. 因此，我们只需证明行的结果. 因为 $\boldsymbol{H}$ 的第一行是

$$1\quad 1\quad 1\quad \cdots\quad 1,$$

而且因为第一行与任意其他行的内积等于 0，所以其他任意行必定有相同数量的 +1 和 −1. 因此，n 是偶数，存在 $(n/2)$ 个 +1 和 $(n/2)$ 个 −1. 交换列，使得在第二行中 +1 在前面，其后是 −1. 这时，前两行有如下形式：

$$\begin{array}{ccccccc}1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1\\ 1 & 1 & \cdots & 1 & -1 & -1 & \cdots & -1\end{array}$$

596

这种交换不影响矩阵的大小及每一行中 +1 的数量，以及两行同时有 +1 的列的数量. 考虑第 i 行，其中 $i\neq 1$, 2. 第 i 行的前半部分(前 $n/2$ 个项)有 u 个 +1 和 $n/2-u$ 个 −1，在后半部分有 v 个 +1 和 $n/2-v$ 个 −1. 第一行和第 i 行的内积等于 0. 因此，第 i 行有 $(n/2)$ 个 +1，且

$$u+v=\frac{n}{2}. \tag{10.17}$$

$i\neq 1$, 2 时，第二行和第 i 行的内积等于 0. 因此

$$u-\left(\frac{n}{2}-u\right)-v+\left(\frac{n}{2}-v\right)=0 \tag{10.18}$$

(参看图 10.2). 因此

$$u-v=0. \tag{10.19}$$

第二行	1	1	−1	−1
第 i 行, $i\neq 1,2$	1	−1	1	−1
	u	$\frac{n}{2}-u$	v	$\frac{n}{2}-v$
	位数	位数	位数	位数

图 10.2　计算规范阿达马矩阵的第二行与第 i 行($i\neq 1$, 2)的内积的示意图

方程(10.17)和(10.19)给出下面的结果：

$$2u=\frac{n}{2}$$

⊖ 本小节余下部分可以跳过.

或

$$n = 4u.$$

因此，n是4的倍数，这就证明了定理10.9的第一个部分．另外，除第一行外，其余各行有$(n/2=2u)$个$+1$，这就证明了定理10.9的另一个部分．最后，对于$i\neq 1$，2，第二行和第i行正好有u个列同时为$+1$．对于行j和行i，其中$j\neq i$，j，$i\neq 1$，通过重复上面的证明，交换列使得在第j行上$+1$在最前面，然后是-1，同样可以证明上述结果成立．这就完成了定理10.9的证明． ■

10.5.3 最丰富的(n, d)代码

下面我们考虑这样的问题：由阿达马矩阵得到的代码是如何"丰富"？即对于同样的n(代码字的长度)和同样的d(代码字之间的最小距离)，与其他可能的代码相比较，这样的代码是有较多还是较少的代码字？ [597]

设$A(n, d)$是(n, d)代码中代码字的最大数量．定理10.3的方程(10.5)给出$A(n, d)$的一个上界．下面的结果给出大多数情况下的一个更好的上界．

定理10.12(Plotkin[1960]) *假设(n, d)代码有N个代码字．如果$d>n/2$，那么*

$$N\leqslant \frac{2d}{2d-n}.$$

证明[⊖] 构建$N\times n$矩阵$\boldsymbol{M}=(m_{ij})$，其行是代码字．考虑下面的表达式：

$$S=\sum d(u, v), \tag{10.20}$$

其中，取和的范围是代码字集合的所有(无序)字对u，v．我们知道，对于所有u，v，有$d(u, v)\geqslant d$，所以

$$S\geqslant \binom{N}{2}d=\frac{N(N-1)}{2}d. \tag{10.21}$$

设$t_0^{(i)}$是$\boldsymbol{M}$的第i列上0的出现次数，而$t_1^{(i)}$是第i列上1的出现次数．注意

$$S=\sum_{\{i,k\}}\sum_{j}|m_{ij}-m_{kj}|=\sum_{j}\sum_{\{i,k\}}|m_{ij}-m_{kj}|. \tag{10.22}$$

现在，

$$\sum_{\{i,k\}}|m_{ij}-m_{kj}|$$

是行i和k的次数，其中一行在第j列上为1而另一行在j列上为0，这个数量由$t_0^{(j)}t_1^{(j)}$给出．因此，根据(10.22)式，有

$$S=\sum_{j}t_0^{(j)}t_1^{(j)}. \tag{10.23}$$

现在

$$t_1^{(j)}=N-t_0^{(j)},$$

所以

$$t_0^{(j)}t_1^{(j)}=t_0^{(j)}[N-t_0^{(j)}].$$

我们寻找$t_0^{(j)}t_1^{(j)}$的一个上界．设$f(x)$是由下面的公式定义的函数：

$$f(x)=x(N-x),$$

[598]

其中$0\leqslant x\leqslant N$．注意，当$x=N/2$时，$f(x)$最大，而且$f(x)$的最大值是$(N/2)(N/2)=N^2/4$．因此

⊖ 本证明可以跳过．

$$t_0^{(j)} t_1^{(j)} \leqslant \frac{N^2}{4},$$

所以，根据(10.23)式，

$$S \leqslant n \frac{N^2}{4}. \tag{10.24}$$

于是，根据(10.21)式和(10.24)式有

$$\frac{N(N-1)d}{2} \leqslant \frac{nN^2}{4},$$

$$(N-1)d \leqslant \frac{nN}{2},$$

$$N\left(d - \frac{n}{2}\right) \leqslant d.$$

因为 $d > n/2$，所以

$$N \leqslant \frac{d}{d - n/2} = \frac{2d}{2d - n}.$$

■

推论 10.12.1 如果 $d > n/2$，则 $A(n, d) \leqslant \frac{2d}{2d-n}$.

一个秩为 4 的规范阿达马矩阵给出一个 $n=4m-1$，$d=2m$ 的(n, d)代码，且有 $N=4m-1$ 个代码字. 因此

$$A(4m-1, 2m) \geqslant 4m-1. \tag{10.25}$$

根据推论 10.12.1，有

$$A(4m-1, 2m) \leqslant \frac{2(2m)}{2(2m)-(4m-1)} = \frac{4m}{1} = 4m. \tag{10.26}$$

因此，从规范阿达马矩阵得到的代码与最佳可能的代码很接近. 根据代码字的数量，通过加入一个代码字 1 1…1，我们可以得到最佳可能的代码. 加入这个代码字等价于修改我们前面的构造法，不删除秩为 $4m$ 的规范阿达马矩阵的第一行. 这个字与由阿达马矩阵所得的任意代码字的距离是 $2m$，因为任意这样的代码字都有 $2m-1$ 个 1 和 $2m$ 个 0. 因此，我们有 $4m$ 个字的代码，其中每一个字的长度是 $4m-1$，而且两个代码字之间的最小距离等于 $2m$. 这个代码称为$(4m-1)$阿达马码($(4m-1)$-Hadamard code). 所以有

$$A(4m-1, 2m) \geqslant 4m. \tag{10.27}$$

599 等式(10.26)和不等式(10.27)给出下面的定理.

定理 10.13 对于存在秩为 $4m$ 的(规范)[1]阿达马矩阵的所有 m，有

$$A(4m-1, 2m) = 4m.$$

这个界是通过使用$(4m-1)$阿达马码得到的.

设 $B(n, d)$是长度为 n 的字的最大数量，其中每一个字与其他字之间的距离都正好是 d. 显然，$B(n, d) \leqslant A(n, d)$.

推论 10.13.1 对于存在秩为 $4m$ 的规范阿达马矩阵的所有 m，有

$$B(4m-1, 2m) = 4m.$$

证明 我们已构造的代码字都有距离 $2m$. ■

注意，有趣的是，有时可以通过把代码字的长度增加 1 而得到丰富得多的代码. 我们将证明下面的定理.

[1] 回想一下，如果存在秩为 $4m$ 的任意阿达马矩阵，那么一定存在规范阿达马矩阵.

定理 10.14 *对于存在秩为 $4m$ 的规范阿达马矩阵的所有 m，有*

$$A(4m,2m)=8m.$$

这一定理表明，在存在秩为 $4m$ 的(规范)阿达马矩阵的情况下，加入一个数字导致可能的代码字数量成倍增加一倍.

我们首先通过构建有 $8m$ 个代码字的$(4m, 2m)$代码来证明 $A(4m, 2m)\geq 8m$.

从一个 $4m\times 4m$ 规范阿达马矩阵 $\boldsymbol{H}$ 开始，我们已经构建了$(4m-1, 2m-1, m-1)$设计. 设 $\boldsymbol{A}$ 是这一设计的关联矩阵. 使用 $\boldsymbol{A}$ 的行和通过交换 $\boldsymbol{A}$ 的 0 和 1 而得的矩阵 $\boldsymbol{B}$ 的行，我们得到总计 $8m-2$ 个字. $\boldsymbol{A}$ 中两个字间的距离是 $2m$，$\boldsymbol{B}$ 中两个字间的距离是 $2m$，如果 $\boldsymbol{A}$ 中的字在 $\boldsymbol{A}$ 的第 i 行而 $\boldsymbol{B}$ 中的字在 $\boldsymbol{B}$ 的第 j 行，且 $i\neq j$，那么这两个字之间的距离是 $2m-1$，而如果 $\boldsymbol{A}$ 中的字在 $\boldsymbol{A}$ 的第 i 行，$\boldsymbol{B}$ 中的字在 $\boldsymbol{B}$ 的第 i 行，那么这两个字之间的距离是 $4m-1$. 现在，在 $\boldsymbol{A}$ 的字前加上数字 1，而在 $\boldsymbol{B}$ 的字前加上数字 0，我们得到两个字的集合 $\boldsymbol{A}'$ 和 $\boldsymbol{B}'$. $\boldsymbol{A}'$ 的两个字及 $\boldsymbol{B}'$ 的两个字间的距离仍然是 $2m$，而如果 $i\neq j$，$\boldsymbol{A}'$ 的第 i 行的字与 $\boldsymbol{B}'$ 的第 j 行的字之间的距离是 $2m$，如果 $i=j$，则距离是 $4m$. 仍仅有 $8m-2$ 个字. 加入如下 $4m$ 个数字的字：

$$0\ \ 0\ \ \cdots\ \ 0$$

和

$$1\ \ 1\ \ \cdots\ \ 1.$$

现在我们有 $8m$ 个字. 这两个新字之间的距离等于 $4m$. $\boldsymbol{A}'$ 和 $\boldsymbol{B}'$ 中的任意一个字有 $2m$ 个 1 和 $2m$ 个 0[因为 $\boldsymbol{A}$ 的字有 $k=(2m-1)$ 个 1]. 因此，这些新字与 $\boldsymbol{A}'$ 或 $\boldsymbol{B}'$ 中的任意字间的距离是 $2m$. 现在，我们有 $8m$ 个代码字，每一个代码字的长度为 $4m$，且两个这样的代码字之间的最小距离是 $2m$. 这个代码字的集合定义$(4m)$阿达马码(4m-Hadamard code). 这等价于通过下面的方法得到的代码字集合：取秩为 $4m$ 的规范阿达马矩阵，不删除第一行或第一列，考虑下面的矩阵： 600

$$\begin{bmatrix} \boldsymbol{H} \\ -\boldsymbol{H} \end{bmatrix},$$

把 -1 变成 0，并使用结果矩阵中的行作为代码字. 从(10.15)式的规范阿达马矩阵得到的 $m=2$ 时的$(4m)$阿达马码如表 10.1 所示. 这证明了，如果存在秩为 $4m$ 的(规范)阿达马矩阵，那么下式成立：

$$A(4m,2m)\geq 8m. \tag{10.28}$$

表 10.1 由(10.15)的规范阿达马矩阵 $\boldsymbol{H}$ 导出的$(4m)$阿达马码的 $8m=16$ 个代码字

1	1	1	1	1	1	1	1
1	0	1	0	1	0	1	0
1	1	0	0	1	1	0	0
1	0	0	1	1	0	0	1
1	1	1	1	0	0	0	0
1	0	1	0	0	1	0	1
1	1	0	0	0	0	1	1
1	0	0	1	0	1	1	0

由 $\boldsymbol{H}$

0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	1	1	0	0	1	1	0
0	0	0	0	1	1	1	1
0	1	0	1	1	0	1	0
0	0	1	1	1	1	0	0
0	1	1	0	1	0	0	1

由 $-\boldsymbol{H}$

现在，我们证明对于所有 $m>0$，有下式成立：

$$A(4m,2m)\leq 8m, \tag{10.29}$$

这个结果将证明定理 10.14.

定理 10.15 *如果 $0<d<n$，那么 $A(n, d)<2A(n-1, d)$.*

证明 考虑 $A(n, d)$ 个字的(n, d)代码 C. 设 C' 是通过选取 C 中所有从 1 开始的代码字并删

除这些代码字的第一个数字而得到的. 那么C'定义一个$(n-1, d)$代码，所以

$$|C'| \leqslant A(n-1,d).$$

类似地，如果C''是通过选取C中从0开始的所有代码字并删除这些代码字的第一个数得到的，那么有

$$|C''| \leqslant A(n-1,d).$$

因此

601

$$A(n,d) = |C| = |C'| + |C''| \leqslant 2A(n-1,d).$$ ■

现在，根据定理10.15和(10.26)式，(10.29)式成立，因为我们有下式：

$$A(4m,2m) \leqslant 2A(4m-1,2m) \leqslant 2(4m) = 8m.$$

(注意，不能直接运用定理10.13来得到这一结果. 为什么?)

我们构建的$(4m)$阿达马码还称为(第一类)Reed-Muller码(Reed-Muller code)，这是以Reed[1954]和Muller[1954]的名字命名的. 定理10.13和定理10.14的结果归功于Levenshtein[1964]，他还得到了更一般的结果.

在结束本小节之际，我们应该注意到，正如我们在构造法中所示，如果交换0和1且对于某个p，$m=2^p$，那么$(4m-1)$阿达马码和$(4m)$阿达马码实际上是线性代码. 然而，24阿达马码就是非线性代码的例子. 在这里，我们不准备给出这些事实的证明，而是提示读者参考编码理论文献：例如，不妨参见MacWilliams和Sloane[1983](还可以参考练习25～练习28).

10.5.4 一些应用

【例10.8 Mariner9空间探测器】 错误校正码已广泛应用于太空行动，如1977年国家航空航天局的探险者号火星的飞行任务. Reed-Muller码早在1972年就被用于Mariner9航天探测器，这一探测器把火星的照片送回地球. 该探测器所使用的特殊代码是$m=8$时的$(4m)$阿达马码，即基于32×32规范阿达马矩阵的代码. 这个代码有64个32位的代码字. 火星的照片被分割成非常小的点，每一个点指定64个层次灰度中的1个层次，而这个层次被编码成一个32位代码字. (关于更详细的内容可以参见Posner[1968]、van LInt[1982]、MacWilliams和Sloane[1983].) ■

【例10.9 错误校正码在光盘设计中的作用】 光盘(CD)使用高"信息密度"对信号进行数字编码. 它被设计为具有强大的错误校正性能，这些性能为CD录制在音响上引发巨大改进提供了重要的基础. 错误可能是由于使用中的磨损、灰尘、制造缺陷等原因引起的. 这样的错误导致"暴音"或连续错误序列. 称为CIRC(Cross-Interleaved Reed-Solomon Code)的编码技术通过使用两个隔行扫描代码能够校正至多4000个连续错误. 关于CD中错误校正码的使用的更详细内容可以参见Bossert等[1977]、Immink[1991]、Peek[1985]、Pinch[2001]、Vries和Odaka[1982]. ■

【例10.10 "解读"DNA到生产蛋白质】 Golomb[1962]推测错误校正码可以用于基因编码中，特别地，在为了生产蛋白质而要"解读"DNA链的过程中要使用这一代码. 当三个基的序列不对应一个氨基酸的代码时，我们可以使用错误校正(参见例2.2). Golomb推测要使用$(4m)$阿达马码. 可以编码20种不同氨基酸的这种最小的代码有24个代码字，这种最小的代码是由一个12×12规范阿达马矩阵构建的(这种矩阵的存在不是从我们的定理得出的). 代码字的长度为12.

602

可以用两个步骤把DNA链编码成0和1的串. 首先，把字母A，C，G和T中的一个字母编码成00，一个字母编码成01，一个编码成10，一个编码成11. 这样就把一个长度为m的DNA链表示成为一个长度为$2m$的位串，即信息字. 把每一个信息字分割成长度为$k=2$的区组且使用$2\to12$代码编码. 这样的信息字的每6个位(链中的每三个字母)，或这个代码的每36个位将对应于一种氨基酸. ■

本节练习

1. 一个阿达马设计包含下面的区组，确定没有给出的区组：

$$\{1,2,4\},\{2,3,5\},\{1,3,6\},\{3,4,7\}.$$

2. 假设 $\boldsymbol{A}$ 是一个 $(b,\ v,\ r,\ k,\ \lambda)$ 设计的关联矩阵，而且假设代码是由这个矩阵的行组成的.
   (a)这一代码中两个字之间的距离是多少？
   (b)这个代码可以检测出多少个错误？
   (c)这一代码可以校正多少个错误？
   (d)假设 $\boldsymbol{B}$ 是补设计的关联矩阵(9.4 节练习 20). 如果 $i\neq j$，$\boldsymbol{A}$ 的第 i 行和 $\boldsymbol{B}$ 的第 j 行间的距离是多少？
   (e)如果 $\boldsymbol{B}$ 的行形成一个代码，那么这个代码可以检测出多少个错误？
   (f)这个代码可以校正多少个错误？
3. 给定一个 9 变元的施泰纳三元系统，通过使用关联矩阵的行构建一个代码. 确定两个代码字间的最小距离.
4. (a)存在 3×3 的阿达马矩阵吗？
   (b)存在 6×6 的阿达马矩阵吗？
5. 如果 $\boldsymbol{H}$ 是阿达马矩阵，把它的某一行乘以 -1，其结果矩阵仍然是阿达马矩阵吗？为什么？
6. 证明：如果存在秩为 $4m$ 的阿达马矩阵，那么一定存在秩为 $4m$ 的规范阿达马矩阵.
7. 假设有下面的矩阵 $\boldsymbol{H}$：

$$\boldsymbol{H}=\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & 1 & -1 & -1\\ 1 & -1 & 1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}.$$

   使用 $\boldsymbol{H}$ 确定一个 8×8 的阿达马矩阵和一个(7，3，1)设计.
8. 从一个 16×16 的规范阿达马矩阵推导出表 9.27 所示的(15，7，3)设计.　603
9. (a)对于 $m=4$，确定一个 $(4m-1)$ 阿达马码.
   (b)这一代码可以检测出多少个错误？
   (c)这一代码又能校正多少个错误？
10. (a)对于 $m=4$，确定一个 $(4m)$ 阿达马码.
    (b)这一代码可以检测出多少个错误？
    (c)这一代码又可以校正多少个错误？
11. 对于 $m=8$，重复练习 9.
12. 对于 $m=8$，重复练习 10.
13. 下面是一个(11，5，2)设计. 使用这一设计确定可以检测多至 5 个错误的 12 个字的代码和可以检测多至 5 个错误的 24 个字的代码.

$$\{1,2,3,4,9\},\{1,2,5,6,10\},\{2,3,6,7,11\},\{1,4,6,7,8\},$$
$$\{2,5,7,8,9\},\{3,6,8,9,10\},\{1,7,9,10,11\},\{2,4,8,10,11\},$$
$$\{4,5,6,9,11\},\{3,4,5,7,10\},\{1,3,5,8,11\}.$$

14. 完成定理 10.10 证明中构建的矩阵 $\boldsymbol{H}$ 定义阿达马矩阵的证明.
15. 从(7，3，1)设计的关联矩阵开始并构建相应的阿达马矩阵来说明定理 10.10 证明中的构造法.
16. 假设 $\boldsymbol{H}$ 是一个 $(4m-1,\ 2m-1,\ m-1)$ 设计的关联矩阵，而且 $\boldsymbol{K}$ 是其补设计的关联矩阵(9.4 节练习 20).
    (a) $\boldsymbol{H}$ 的第 i 行与 $\boldsymbol{H}$ 的第 j 行之间的距离是多少？
    (b) $\boldsymbol{H}$ 的第 i 行与 $\boldsymbol{K}$ 的第 j 行之间的距离是多少？
17. 假设 S 是长度为 n 的二进制代码字的集合，且 S 中每一对字 a 和 b 的汉明距离 $d(a,\ b)$ 为 d. 设 T 定义为 S 中的字的补(交换 0 和 1)的集合.
    (a) T 的两个字之间的距离是多少？

(b) S 的字和 T 的字之间的距离是多少?

(c)如果 $n=12$, $d=5$, 那么代码 T 可以检测出多少个错误? 它又能校正多少错误?

(d)假设从 S 中的字和 T 中的字定义一个代码. 如果 $n=12$, $d=5$, 那么这个代码将检测出多少个错误? 它又能校正多少错误?

18. 假设 $m\times n$ 矩阵 $\boldsymbol{M}$ 是一个区组设计的关联矩阵，且第 i 行和第 j 行在 u 个列上同时为 1.

(a)这两行的内积是多少?

(b)在其补设计中内积是多少(参见 9.4 节练习 20)?

604

19. 确定 1 和 -1 构成的 2×4 矩阵，使得如果 $i\neq j$ 则行 i 和 j 的内积等于 0，如果 $i=j$，则内积等于 4.

20. 对于 k 的什么值，存在 1 和 -1 构成的 $2\times k$ 矩阵，使得如果 $i\neq j$ 则行 i 与 j 的内积等于 0，如果 $i=j$，则内积等于 k?

21. 证明当 n 足够大时，$A(n, n-1)=2$.

22. (a)证明如果 $n>m$，那么 $A(n, d)\geqslant A(m, d)$.

(b)如果 $n>m$，是否一定有 $A(n, d)>A(m, d)$? 为什么?

23. (a)确定 $A(4m-2, 2m)$的一个上界.

(b)对于任意大的值 m，正确地确定 $A(4m-2, 2m)$.

24. 使用练习 17 的结果，证明:

$$A(2d,d)\geqslant 2B(2d,d).$$

25. 证明我们定义的$(4m-1)$阿达马码不是线性代码. [提示：考虑代码字的和.]

26. 如果 $m=2^p$，且在我们定义的$(4m-1)$阿达马码中交换 0 和 1，那么这个代码变成线性代码. 通过寻找生成矩阵，证明 $m=1$ 时这一结论成立.

27. 继续练习 26，如下证明当 $m=2$ 时的线性性.

(a)使用取下面矩阵生成相应代码而得到的生成矩阵

$$\mathbf{G}=\begin{bmatrix}0&1&1&1\\1&0&1&1\\1&1&0&1\end{bmatrix}.$$

(b)证明这个代码与(10.4 节练习 25)$m=2$ 且 0 和 1 已交换时的$(4m-1)$阿达马码等价.

28. 如果 $m\neq 2^p$，证明即使交换$(4m-1)$阿达马码中的 0 和 1，所得的代码也不是线性的[提示：一个线性代码有多少个代码字].

29. 回想一下，一个位串 x 的汉明权(Hamming weight)，记为 $wt(x)$，是 $\boldsymbol{x}$ 中非零数字的数量. 设 $A(n, d, w)$是(n, d)代码中每一个字有相同汉明权 w 的代码字的最大数量. 证明:

(a)$A(n, 2d-1, w)=A(n, 2d, w)$　　(b)$A(n, 2d, w)=A(n, 2d, n-w)$

(c)$A(n, 2d, w)=1$，如果 $w<d$　　(d)$A(n, 2d, d)=\lfloor n/d\rfloor$

参考文献

Avizenius, A., "Fault-Tolerant Systems," *IEEE Trans. Comput., C-25* (1976), 1304–1312.

Berlekamp, E. R., *Algebraic Coding Theory*, McGraw-Hill, New York, 1968.

Blake, I. F., and Mullin, R. C., *The Mathematical Theory of Coding*, Academic Press, New York, 1975. (Abridged as *An Introduction to Algebraic and Combinatorial Coding Theory*, Academic Press, New York, 1976.)

Bock, H. H. (ed.), *Classification and Related Methods of Data Analysis*, North-Holland, Amsterdam, 1988.

605

Bose, R. C., and Shrikhande, S. S., "A Note on a Result in the Theory of Code Construction," *Inf. Control, 2* (1959), 183–194.

Bossert, M., Braĭtbakh, M., Zyablov, V. V., and Sidorenko, V. R., "Codes That Correct Multiple Burst Errors or Erasures (Russian)" *Problemy Peredachi Infor-*

matsii, 33 (1997), 15–25; translation in *Problems Inform. Transmission, 33* (1997), 297–306.

CAMERON, P. J., and VAN LINT, J. H., *Designs, Graphs, Codes, and Their Links*, London Mathematical Society Student Texts 22, Cambridge University Press, Cambridge, 1991.

CARTER, W. C., and BOURICIUS, W. G., "A Survey of Fault-Tolerant Architecture and Its Evaluation," *Computer, 4* (1971), 9–16.

CRAIGEN, R., and WALLIS, W. D., "Hadamard Matrices: 1893 - 1993," *Congr. Numer., 97* (1993), 99–129.

DAY, W. H. E., "The Sequence Analysis and Comparison Bibliography," at `http://www.classification-society.org/sequence.html`, Oct. 12, 2002.

DAY, W. H. E., and MCMORRIS, F. R., "Critical Comparison of Consensus Methods for Molecular Sequences," *Nucleic Acids Research, 20* (1992), 1093–1099.

DAY, W. H. E., and MCMORRIS, F. R., "Discovering Consensus Molecular Sequences," in O. Opitz, B. Lausen, and R. Klar (eds.), *Information and Classification: Proceedings of the 16th Annual Conference of the Gesellsshaft fuer Klassifikation e.V.*, Springer-Verlag, Heidelberg, 1993, 393–402.

DAY, W. H. E., and MCMORRIS, F. R., *Axiomatic Consensus Theory in Group Choice and Biomathematics*, SIAM Publications, Philadelphia, 2003.

DORNHOFF, L. L., and HOHN, F. E., *Applied Modern Algebra*, Macmillan, New York, 1978.

FISHER, J. L., *Application-Oriented Algebra: An Introduction to Discrete Mathematics*, Harper & Row, New York, 1977.

GALAS, D. J., EGGERT, M., and WATERMAN, M. S., "Rigorous Pattern-Recognition Methods for DNA Sequences. Analysis of Promoter Sequences from *Escherichia Coli*," *J. Molecular Biol., 186* (1985), 117–128.

GOLAY, M. J. E., "Notes on Digital Coding," *Proc. IEEE, 37* (1949), 657.

GOLDIE, C. M., and PINCH, R. G. E., *Communication Theory*, London Mathematical Society Student Texts 20, Cambridge University Press, Cambridge, 1991.

GOLOMB, S. W., "Efficient Coding for the Deoxyribonucleic Channel," in *Mathematical Problems in the Biological Sciences*, Proceedings of Symposia in Applied Mathematics, Vol. 14, American Mathematical Society, Providence, RI, 1962, 87–100.

HADAMARD, J., "Résolution d'une Question Relative aux Déterminants," *Bull. Sci. Math., 17* (1893), 240–248.

HALL, M., *Combinatorial Theory*, Ginn (Blaisdell), Boston, 1967. (Second printing, Wiley, New York, 1980.)

HAMMING, R. W., "Error Detecting and Error Correcting Codes," *Bell Syst. Tech. J., 29* (1950), 147–160.

HILL, R., *A First Course in Coding Theory*, Oxford University Press, New York, 1986.

IMMINK, K. A. S., *Coding Techniques for Digital Recorders*, Prentice Hall International, Hertfordshire, UK, 1991.

JANOWITZ, M., LAPOINTE, F.-J., MCMORRIS, F. R., MIRKIN, B., and ROBERTS, F. S., *Bioconsensus*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 61, American Mathematical Society, Providence, RI, 2003.

JOHNSON, P. E., *Social Choice: Theory and Research*, Sage Publications, Thousand Oaks, CA, 1998. 606

KANNAN, S., "A Survey of Tree Consensus Criteria and Methods," in S. Tavare (ed.), *Proceedings of Phylogeny Workshop*, DIMACS Tech. Rep. 95–48, DIMACS Center, Rutgers University, Piscataway, NJ, 1995.

LEVENSHTEIN, V. I., "The Application of Hadamard Matrices to a Problem in Coding," *Probl. Kibern., 5* (1961), 123–136. [English transl.: *Probl. Cybern., 5* (1964), 166–184.]

MACWILLIAMS, F. J., and SLOANE, N. J. A., *The Theory of Error-Correcting Codes*, Vols. 1 and 2, North-Holland, Amsterdam, 1983.

MIRKIN, B., and ROBERTS, F. S., "Consensus Functions and Patterns in Molecular Sequences," *Bull. Math. Biol.*, *55* (1993), 695–713.

MULLER, D. E., "Application of Boolean Algebra to Switching Circuit Design and to Error Detection," *IEEE Trans. Comput.*, *3* (1954), 6–12.

PEEK, J. B. H., "Communications Aspects of the Compact Disc Digital Audio System," *IEEE Commun. Magazine*, *23* (1985), 7–15.

PETERSON, W. W., *Error Correcting Codes*, MIT Press, Cambridge, MA, 1961.

PETERSON, W. W., and WELDON, E. J., *Error Correcting Codes*, MIT Press, Cambridge, MA, 1972. (Second edition of Peterson [1961].)

PINCH, R. G. E., "Coding Theory: The First 50 Years," at **http://pass.maths.org/issue3/codes/**, Feb. 21, 2001.

PLESS, V., *Introduction to the Theory of Error-Correcting Codes*, Wiley, New York, 1998.

PLOTKIN, M., "Binary Codes with Specified Minimum Distances," *IEEE Trans. Inf. Theory*, *6* (1960), 445–450.

POLI, A., and HUGUET, L., *Error Correcting Codes: Theory and Applications*, Prentice Hall International, Hemel Hempstead/Masson, Paris, 1992.

POSNER, E. C., "Combinatorial Structures in Planetary Reconnaissance," in H. B. Mann (ed.), *Error Correcting Codes*, Wiley, New York, 1968.

REED, I. S., "A Class of Multiple-Error-Correcting Codes and the Decoding Scheme," *IEEE Trans. Inf. Theory*, *4* (1954), 38–49.

ROBERTS, F. S., *Discrete Mathematical Models, with Applications to Social, Biological, and Environmental Problems*, Prentice Hall, Englewood Cliffs, NJ, 1976.

SELLERS, F. F., HSIAO, M. Y., and BEARNSON, L. W., *Error Detecting Logic for Digital Computers*, McGraw-Hill, New York, 1968.

SLEPIAN, D., "A Class of Binary Signaling Alphabets," *Bell Syst. Tech. J.*, *35* (1956), 203–234. (a)

SLEPIAN, D., "A Note on Two Binary Signaling Alphabets," *IEEE Trans. Inf. Theory*, *2* (1956), 84–86. (b)

SLEPIAN, D., "Some Further Theory of Group Codes," *Bell Syst. Tech. J.*, *39* (1960), 1219–1252.

SYLVESTER, J. J., "Thoughts on Inverse Orthogonal Matrices, Simultaneous Sign Successions, and Tesselated Pavements in Two or More Colors, with Applications to Newton's Rule, Ornamental Tile-Work, and the Theory of Numbers," *Philos. Mag.*, *34* (1867), 461–475.

TIETÄVÄINEN, A., "On the Nonexistence of Perfect Codes over Finite Fields," *SIAM J. Appl. Math.*, *24* (1973), 88–96.

VAN LINT, J. H., "Nonexistence Theorems for Perfect Error-Correcting Codes," in *Computers in Algebra and Number Theory*, Vol. 4 (SIAM-AMS Proceedings), 1971.

607

VAN LINT, J. H., "Coding, Decoding and Combinatorics," in R. J. Wilson (ed.), *Applications of Combinatorics*, Shiva, Nantwich, UK, 1982, 67–74.

VAN LINT, J. H., *Introduction to Coding Theory*, 3rd ed., Springer-Verlag, New York, 1999.

VRIES, L. B., and ODAKA, K., "CIRC - The Error-Correcting Code for the Compact Disc Digital Audio System," *Collected Papers from the AES Premiere Conference, Rye, New York*, Audio Engineering Society, New York, 1982, 178–188.

WAKERLY, J., *Error Detecting Codes, Self-Checking Circuits, and Applications*, Elsevier North-Holland, New York, 1978.

WATERMAN, M. S., "Consensus Patterns in Sequences," in M. S. Waterman (ed.), *Mathematical Methods for DNA Sequences*, CRC Press, Boca Raton, FL, 1989, 93–115.

WATERMAN, M. S., *Introduction to Computational Biology; Maps, Sequences and Genomes*, CRC Press, Boca Raton, FL, 1995.

WATERMAN, M. S., ARRATIA, R., and GALAS, D. J., "Pattern Recognition in Several Sequences: Consensus and Alignment," *Bull. Math. Biol.*, *46* (1984), 515–527.

608

WELSH, D., *Codes and Cryptography*, Oxford University Press, New York, 1988.

第 11 章　图论中的存在问题

图论中的很多课题构成组合数学的大部分基础. 本章以及后续的两章主要研究图和网络. 图论的现代方法深受计算机科学的影响，而且它强调用于解决问题的算法. 在这些章节中，我们对其中的某些图算法给予介绍. 我们还要讨论图论的很多应用，其中一些已在 3.1 节中介绍过了. 本章着重讨论图论中的存在问题，而第 13 章着重强调优化问题. 第 12 章是一个过渡章，它开始于存在问题而结束于优化问题. 当然，很难做出严格的划分. 对于存在问题所使用的图论概念和技术通常对优化问题也同样有用，反之亦然.

本章验证四个基本存在问题并讨论它们的应用. 这些存在问题是：

1. 给定的图 G 是连通的吗? 即对于 G 中的每对顶点 x，y，一定存在 x 和 y 之间的链吗?
2. 给定的图 G 有强连通方向吗?
3. 给定的图 G 有欧拉链吗，即每一条边正好使用一次的链?
4. 给定的图 G 有哈密顿链吗，即每一个顶点正好使用一次的链?

我们给出若干帮助我们回答这些问题的定理. 考虑实际应用，我们将集中精力于陈述回答这些问题的好过程或算法. 这些结果将应用于诸如运输流量、RNA 代码、街道清扫和电信等方面.

11.1　深度优先搜索：连通性测试

假设 $G=(V, E)$ 是图. 我们要问的第一个问题是：G 是连通的吗? 即对于 G 中每一对顶点 x，y，存在 x，y 之间的链吗?

给定一个小图，从这个图所对应的图示很容易看出它是否是连通的. 然而，对于较大的图来说，画出这样的图示是不可能的. 另外，作为计算机的输入图示不适合于使用. 而你必须通过列出图的边把它输入到计算机中(关于更多如何把图输入到计算机中的方法的讨论参见 3.7 节). 无论是哪种情况，现在都不能很容易地测试一个图是否连通. 因此，我们想要一种测试连通性的算法. 在 11.1.1 节，我们提出一个这样的算法. 这个算法对求解 11.2 节中研究的运输流量问题起着重要的作用，而且还可以利用这一算法寻找欧拉链，这将是我们在 11.3 节中所研究的问题. 这一算法不局限于存在问题，也对解决图论中的优化问题起着重要的作用. 然而，对于巨大的图，即当边的集合不适合计算机的可用 RAM(Random access memory)时，我们就需要全新的算法. 我们将在 11.1.4 节对这一问题进行更深入的讨论.

11.1.1　深度优先搜索

假设 $G=(V, E)$ 是任意图. 我们描述的测试 G 是否连通的方法基于*深度优先搜索过程*(depth-first search procedure)，这一方法是很多计算机算法的基础，是一个高效的过程(关于更详细的讨论可以参见 Aho，Hopcroft 和 Ullman[1974，Ch. 5]、Baase[1992]、Cormen，Leiserson 和 Rivest[1999]、Golumbic[1980]、Hopcroft 和 Tarjan[1973]、Reingold，Nievergelt 和 Deo[1977]、Tarjan[1972]，有关另一个相关的称为广度优先搜索的过程可以参见练习 9).

在深度优先搜索的过程中，我们从一个有 n 个顶点的图 G 开始，目标是用整数 $1, 2, \cdots, n$ 标记这些顶点. 我们选择任意一个顶点并将其标记为 1. 在用整数 k 标记给定的顶点 x 之后，接下来我们在 x 的所有邻居之间搜索，寻找一个未标记的邻居，比如说它是 y. 我们给顶点 y 下一

个标签$k+1$. 我们还通过标记(marking)从x到y经过的边$\{x, y\}$来记录标签过程中所使用的边. 接着我们继续在y的邻居之间搜索. 使用这一方法，我们沿着离开x的G的一条链前进. 如果我们标记了一个顶点z，而它没有未标记的邻居，那么将会出现复杂的情况. 这时，我们返回到z被标记时的邻居u，从u继续搜索，u称为z的双亲(parent). 因为我们标记了一个顶点到下一个标记顶点的边，所以我们可以记录一个顶点的双亲. 这一过程持续进行，直到所有标签1，2，…，n都已用尽(等价于所有顶点已被标记完毕)，或由于我们已返回到没有双亲的已标记顶点，即标签1的顶点而不可能继续进行这一过程为止.

610

我们用图11.1中的图说明这一标签过程. 首先选择顶点a，将其标记为1. 图中数字所示的标签用圆圈圈起来写在各顶点的旁边. 顶点b是顶点a的未标记的邻居，它是下一个被选中的顶点，被标记为2. 边$\{a, b\}$被标记出来(在图中我们画了一个对号). 接下来，寻找顶点b的未标记的邻居，并标记为3. 这就是顶点c. 边$\{b, c\}$被标记. c的未标记的邻居顶点e被标记为4，而边$\{c, e\}$被标记. e的未标记的邻居顶点d被标记为5，而边$\{e, d\}$被标记. 现在，在顶点e被标记为5之后，d的所有邻居都已经被标记. 因此，我们返回e，即d的双亲，并寻找一个未标记的邻居. 我们发现顶点f，把它标记为6. 我们也标记边$\{e, f\}$. 现在，我们用整数7标记顶点h，并标记边$\{f, h\}$，用整数8标记顶点g，并标记边$\{h, g\}$. 现在g的所有邻居都已经被标记，所以我们返回到g的双亲，即顶点h. h的所有邻居都已经被标记，所以我们返回到h的双亲，即顶点f. 最后，因为f的所有邻居已经被标记，所以我们返回f的双亲，即顶点e. 从e出发，我们接下来用整数9标记顶点j，并标记边$\{e, j\}$. 于是，从j出发，我们用整数10标记顶点i，并标记边$\{j, i\}$. 现在，所有的顶点都已经被标记，所以我们停止.

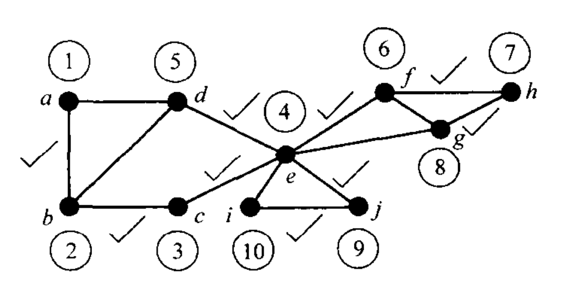


图11.1　使用深度优先搜索过程而得到的一个顶点标签. 对号表示标记边

容易证明，标签过程可以完成当且仅当这个图是连通的. 因此，深度优先搜索过程为测试一个图是否连通提供了一种算法. 图11.2给出对应非连通图的标签过程. 使用标签4标记顶点d之后，我们发现d的双亲，即顶点c已经没有未标记的邻居. 另外，c的双亲，即顶点a也是如此. 最后，a没有双亲，所以我们不能继续标签过程.

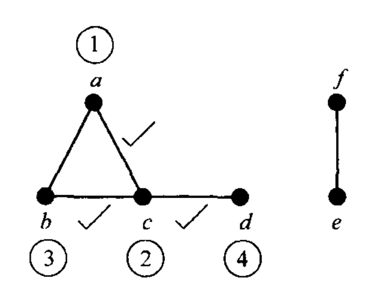


图11.2　一个非连通图的深度优先搜索标签、标记过程

还应该注意的是，如果标签过程完成，而且T是已标记的边的集合，那么T中存在$n-1$个边，因为每向T中加入一个边我们就指定一个新的标签. 另外，假设H是G的支撑子图，它的边都是T中的边. 那么H没有回路，因为加入到T中的每一条边是从标记为i的顶点到标记为j的顶点的，且$i<j$. 因此，H是G的子图，$n=e+1$且没有回路. 根据定理3.20，H是树. 它称为深度优先搜索支撑树(depth-first search spanning tree). (读者应该注意到，图11.1的标记边形成一棵树.)

611

11.1.2　深度优先搜索的计算复杂度

这一深度优先搜索算法的计算复杂度可以按如下方式计算. 假设这个图有n个顶点和e条边. 这个过程的每一个步骤都涉及指定一个标签或经过一条边. 这种经过或者向前从一个标签顶点到一个邻居，或向后从一个顶点到它的双亲. 通过不仅标记用于向前方向的边，而且还标记

在向前搜索中被查访且通向已标签顶点的边，我们可以肯定向前方向的任意一条边都只被使用或查访一次. 不难看出，向后搜索所使用的边同样不超过一次. 因此，整个过程至多查访 $2e$ 条边. 因为至多指定 n 个标签，这个过程至多以 $n+2e$ 步骤终止[用 2.18 节的表记法，这个算法是 $O(n+e)$]. 因为 $2e$ 至多是

$$2\binom{n}{2}=2\,\frac{n(n-1)}{2}=n^2-n.$$

这是一个用顶点的数量表示的关于复杂度的多项式边界[这是一个 $O(n^2)$ 算法㊀]. 这个算法效率很高.

应该注意，可以为有向图定义类似的有向深度优先搜索过程. 同前面一样，我们简单地标记，但是只能从给定顶点到达通过弧从这个顶点可达到的顶点. 这一方法可以用于提供强连通性的高效测试. 对于细节，读者可以参考 Aho，Hopcroft 和 Ullman[1974，Ch. 5]、Cormen，Leiserson 和 Rivest[1999]、Hopcroft 和 Tarjan[1973]，Tarjan[1972]. (还可以参见练习 8).

11.1.3 深度优先搜索算法的形式描述㊁

对于熟悉递归编程的读者，我们给出深度优先搜索算法的更形式的描述来结束本节. 首先，我们描述一个称为 DFSEARCH(v, u)的子程序. 在这一子程序中，以及在其主算法中，k 代表指定标签的当前值，而 T 是已标记边的集合. (我们忽略被查访但通向已标签顶点的边的标记.) 612 控制是当前正在搜索邻居的顶点. 有些顶点已经拥有标签. 顶点 v 是正要被标记的顶点，u 是 v 的双亲.

算法 11.1：DFSEARCH(v, u)

输入： 一个图 G，它的某些顶点已被标记而某些边在 T 中，正要被标记的顶点 v，顶点 v 的双亲顶点 u，以及将指定的当前标签 k.

输出： 某些额外的标签分配，(可能是)新的集合 T 和新的当前标签 k.

步骤 1. 设置控制为顶点 v，标记 v 为已标签，把标签 k 指定给 v，并用 $k+1$ 取代 k.

步骤 2. 对于 v 的每一个邻居 w，如果 w 没有被标记，则把边 $\{v, w\}$ 加入到 T，标记边 $\{v, w\}$，称 v 是 w 的双亲，执行算法 DFSEARCH(w, v).

步骤 3. 如果 v 没有更多未标记的邻居，而且所有顶点已经被标记，停止并输出标签. 如果某个顶点仍没有被标记，那么设置控制为 u 并停止.

现在，我们可以如下概括完整深度优先搜索算法.

算法 11.2：深度优先搜索

输入： 一个有 n 个顶点的图 G.

输出： 使用整数 1，2，…，n 的图 G 的顶点标签，及把 $n-1$ 条 G 的边分配给一个集合 T，或者过程不能完成的信息.

步骤 1. 设置 $T=\varnothing$，$k=1$，并设没有顶点被标记.

步骤 2. 挑选任意一个顶点 v，引入一个(虚设)顶点 α，称 α 是 v 的双亲，并执行 DFSEARCH(v, α).

㊀ 某些作者称这一算法是线性的，因为它的复杂度对于 $n+e$ 是线性的. 其他作者称它为二次的，因为它的复杂度是 n 的平方.

㊁ 本小节可以跳过.

步骤 3. 如果控制被设为 α，那么输出这一过程不能完成的信息.

算法 11.2 或者因为所有顶点都已被标记而终止，或者因为标签过程返回到标记为 1 的顶点，且这个顶点的所有邻居都已有标签，且存在某些未标记的顶点而终止. 对于后者的情况，容易看出这个图是非连通的.

11.1.4 巨大图的连通性测试⊖

涉及电信业务量和网络数据等现代应用中产生的图可能是非常巨大的，以至于它们的边的集合不适合现代计算机的主内存量. 例如，庞大的电话呼叫图就是一个这样的图，这一图的顶点是电话号码，而这个图的边是电话号码之间的呼叫.（更一般地，我们把这种呼叫看成是有向的，我们要查看多重有向图.）这样的图可能有几十亿条边. 当边集合不适合 RAM(Random Access Memory)时，大部分经典的图论算法都不再适用.（甚至使用 6gigabytes 的主内存的计算机也不能装下实际使用中所产生的整个电话呼叫图.）这对开发"外部存储器算法"提出要求. 这样的连通性算法的例子可以在 Aho，Buchsbaum 和 Westbrook[2002]中找到. 其他关于庞大的电话呼叫图和关于因特网的连通性的图的工作，都着重强调这样的图随着时间的变化，以及它们在连通性上的变化. 关于这些课题，可以参见 Aiello，Chun 和 Lu[2000]. 有关外部存储器算法的内容，可以参见 Abello 和 Vitter[1999]. 特别地，可以参见 Abello，Pardalos 和 Resende[1999]的文章.

613

本节练习

1. 对于图 11.3 中的每一个图，从标签为 a 的顶点出发执行深度优先搜索. 指明所有顶点的标签和所有标记边.
2. 表 11.1 给定了顶点为 a，b，c，d，e，f，g 的图 G 的邻接结构，使用深度优先搜索确定 G 是否连通.

表 11.1 图的邻接结构

顶点 x	a	b	c	d	e	f	g
与 x 邻接的顶点	e, d	c, g	b, g	a, e	a, d, f	e	b, c

3. 根据下面图 G 的每一个邻接矩阵，使用深度优先搜索确定 G 是否连通.

$$(a)\ \boldsymbol{A}=\begin{bmatrix}0&1&0&0&0&0&1\\1&0&0&1&1&0&0\\0&0&0&1&0&1&0\\0&1&1&0&0&0&1\\0&1&0&0&0&0&0\\0&0&1&0&0&0&0\\1&0&0&1&0&0&0\end{bmatrix}\qquad (b)\ \boldsymbol{A}=\begin{bmatrix}0&0&0&1&1&0&0&0\\0&0&1&0&0&1&0&0\\0&1&0&0&0&0&0&0\\1&0&0&0&1&0&0&1\\1&0&0&1&0&0&1&0\\0&1&0&0&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&1&0&0&0&0\end{bmatrix}$$

4. 对于图 11.3 中的每一个图，寻找深度优先搜索支撑树.
5. 是否每一棵支撑树都可以作为深度优先搜索支撑树得到？为什么？
6. 假设修改深度优选搜索算法，使得当向前标签不可能，但仍存在未标记顶点时，可选择某个未标记顶点，赋给该顶点一个新标签，并再次开始这个过程. 你能说出由标记边(在 T 中的边)所确定的子图是什么吗？

614

7. 如何修改书中所描述的深度优先搜索算法，使得可以用于计数图 G 的连通分支的数量？
8. (a)下面的有向图 D 的强连通性测试是否正确？选出有向图的任意一个顶点，赋标签 1，并执行有向深

⊖ 本小节可以跳过.

度优先搜索，直到没有顶点可以标记．这时，D 是强连通的当且仅当所有顶点都得到了标签．

(b)如果这不是正确的测试，那么如何修改这个测试？

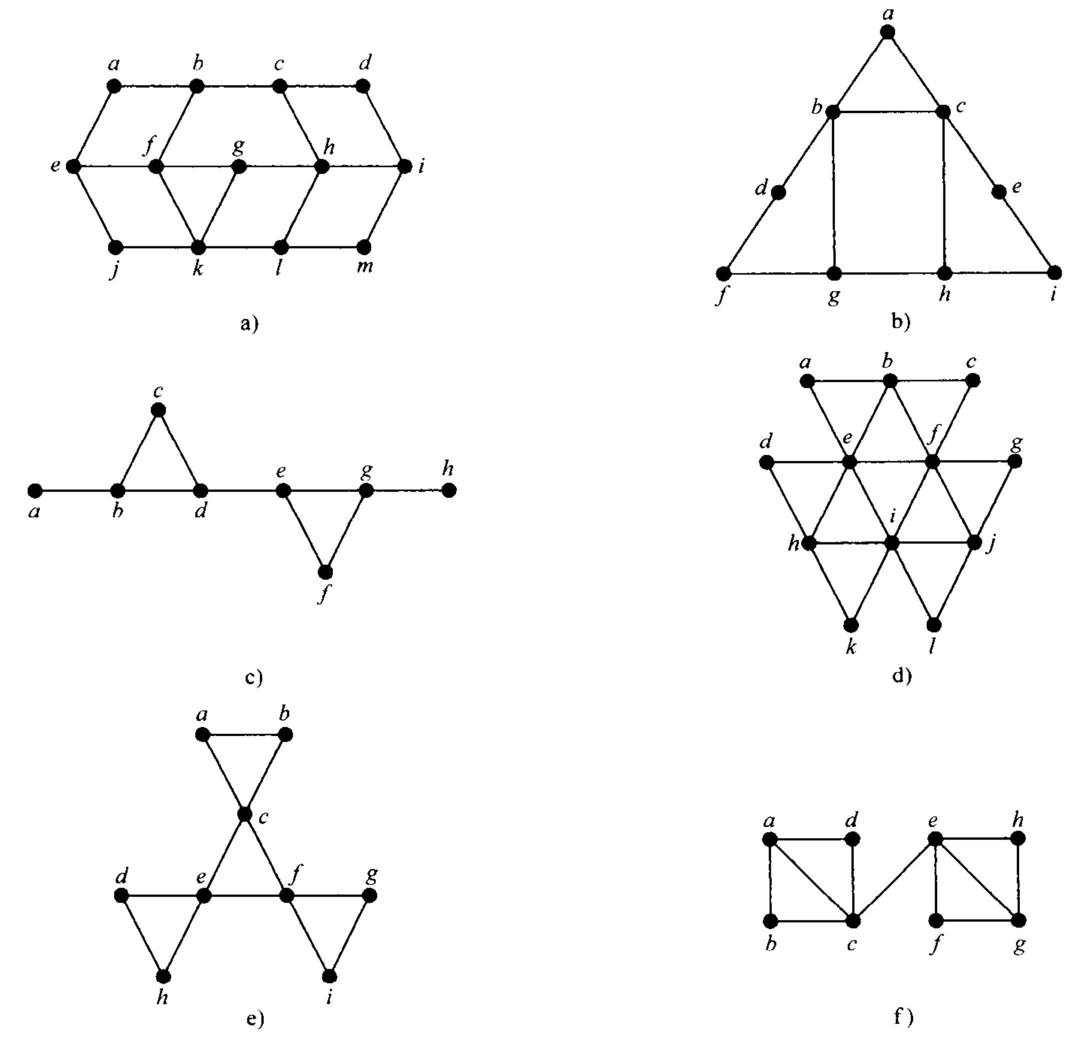


图 11.3　11.1 节练习的图

9. 在广度优先搜索(breadth-first search)算法中，我们从任意选出的顶点 x 出发，把它放置到一个有序列表或队列的前面．在这一算法的每一步骤，我们在这个队列的前面选取一个顶点 y，从这个队列中删除 y，并把与 y 邻接的所有不曾在这个队列中的顶点，逐一加到队列的尾部．我们一直继续下去，直到队列为空．如果所有顶点都不曾在队列中，那么我们从另一个任意选中的不曾在队列中的顶点开始．广度优先搜索从开始的顶点出发在所有方向上延伸，而深度优先搜索从开始的顶点到结束的顶点一次只查访一条链．在 3.3.4 节的算法 3.1 中，我们已经使用了广度优先搜索．关于广度优先搜索的详细讨论可以参见 Baase[1992]、Even[1979]、Golumbic[1980]．对图 11.3 中的每一个图执行广度优先搜索，从顶点 a 开始．按顶点到达队列前面的顺序通过用整数 1 到 n 标记每一个顶点，指明所遇到的顶点的顺序(注意：在很多应用中，与深度优先搜索比较，广度优先搜索效率不高，而且应用也不广泛．例如，当测试给定图是否是平面图的算法时就是这样．然而，在很多网络优化问题中，正如我们将在第 13 章中所研究的大多数问题那样，广度优先搜索是基础过程)．

10. 广度优先搜索(练习 9)可以用于测试图的连通性吗？如果能够测试，如何进行？

11. 广度优先搜索的计算复杂度是什么？

12. 考虑在给定区域代码下，电话号码在 6 周期间的电话呼叫图．

(a)假设所有可能的电话号码都是可用的，那么这个图的顶点集合的大小是多少?
(b)如果每一个电话号码每天平均收到 12 次呼叫，那么这个图的边的集合的大小是多少?

11.2　单行线问题

11.2.1　Robbin 定理

本节讨论图论对于运输流量问题的应用. 想象一下，一座城市有很多位置，其中一些位置是由双行线连接起来的. 由于街道上汽车数量的显著增加，结果造成交通阻塞和空气污染，这表明这座城市应该把所有街道改建成单行线. 这样大概可以缓解交通阻塞. 问题是：是否总能做到这一点呢? 如果不是总能做到，那么什么时候可以做到呢? 答案是肯定的，它总是可以做到的. 只要在每一条街道上放置一个单行信号灯! 然而，如果我们随意地分配信号灯，肯定会遇到麻烦，例如，我们最终完全可能只能从某个地方进入而无法从这个地方出来(例如，参见图 11.4. 这个图给出某座城市的街道方向的一种分配，这种分配只满足某个人的需求，即这个人恰好在位置 a 处有停车场). 我们想把每一条街道都建成单行线，使得对于每一对位置 x 和 y，既可以(合法地)从 y 到达 x，也可以从 x 到达 y. 这总可以做到吗?

615 ~ 616

为了解决这一问题及简便起见，我们假设所有街道在开始时是双行线. 关于某些街道开始时是单行线的这一问题的处理方法，参见 Boesch 和 Tindell[1980]、Roberts[1978](还可参见练习 13). 我们用图表示一座城市的运输网络. 设 G 的顶点表示所讨论的位置，在位置 x 和 y 之间画一条边，当且仅当 x 和 y 由一条双向街道连接. 图 11.5a 给出这样的图的一个简单例子. 使用这样的图，现在可以如下重新陈述我们的问题：能否在图 G 的每一条边上放置一个方向或箭头(单行标志)，使得通过跟踪所产生的图，即有向图中的箭头，总能够从任意一点 x 到达另外任意一点 y. 如果这不总是可能的，什么时候这是可能的? 形式地，当我们给图 G 的每一条边分配方向时，我们定义 G 的*方向*(orientation). 使用 3.2 节的术语，我们寻找强连通图 G 的方向.

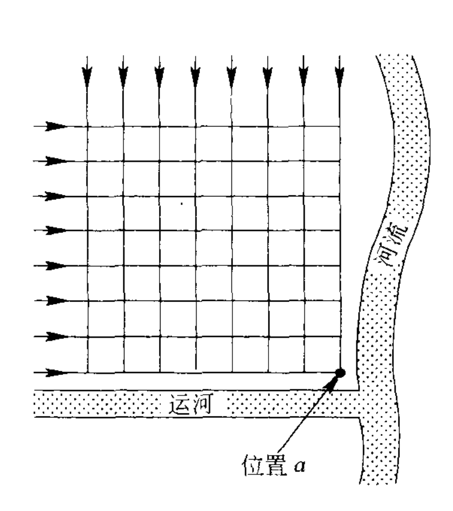


图 11.4　把旅行者钉在位置 a 处的城市街道的单行线分配

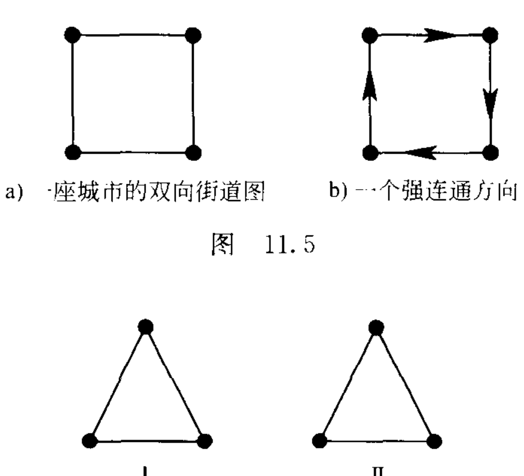


图 11.5

图 11.6　表示非连通城市的图

在图 11.5a 的图中，我们肯定可以给每一条边指定一个方向来得到一个强连通有向图. 图 11.5b 就是我们得到的图. 然而，不是总能够得到强连通的方向. 例如，如果我们的图有两个部分，如图 11.6 中的图所示，那么没有方法给边指定方向，使得从图中部分 I 的顶点可以到达部分 II 的顶点. 为了有一个强连通方向，我们的图必须是连通的. 然而，即使是连通图也不一定有强连通

617

方向. 考虑图 11.7 中的图. 它是连通的. 我们必须在边 α 上设定一个方向. 如果 α 的方向为从部分Ⅰ到部分Ⅱ，那么从Ⅱ中任意顶点都不可能到达部分Ⅰ的任意顶点. 如果 α 的方向是从Ⅱ到Ⅰ的，那么从Ⅰ的任意顶点都不能到达Ⅱ的任意顶点. 这个边 α 的特殊性是什么呢? 答案是，它是唯一连接这个图的两个分离部分的边. 更形式地，删除 α(但不删除边连接的两个顶点)会导致一个非连通图. 连通图中这样的边称为桥(bridge). 图 11.8 给出另一个有桥 α 的例子. 显然，对于有强连通方向的图 G，G 必须不仅是连通的，而且它还必须没有桥.

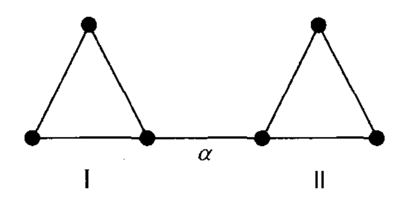


图 11.7　α是一个桥

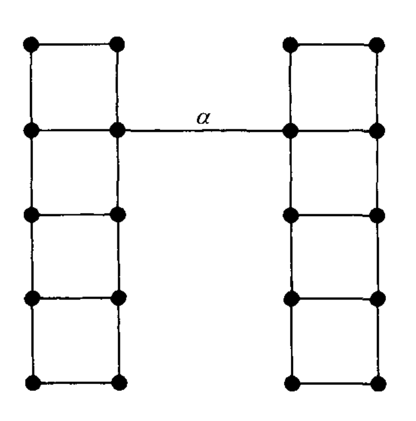


图 11.8　α也是一个桥

在图 11.8 中，假设我们增加另外一个桥 β 来连接由桥 α 接连起来的两个分离的部分，得到图 11.9的图. 这个图有强连通方向吗? 答案是有的. 一个令人满意的分配如图 11.10 所示. 这不是与我们的观察不符吗? 即如果图 G 有强连通方向，那么它不可能有桥. 在这里，答案是否定的. 因为我们不能过早地称 β 为桥. 在图论的意义下，图 11.9 的图中的 α 和 β 都不是桥. 图论的桥有令人讨厌的习性，如果我们有一个桥，并构建另一个桥，那么二者都不是桥. 在应用组合数学中，如果我们使用诸如桥这样的提示性的术语，那么我们必须小心与我们的用法一致.

618

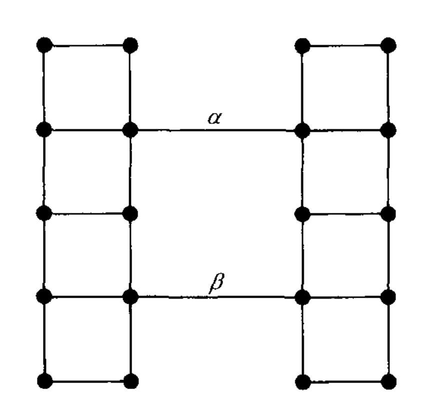


图 11.9　边 α 不再是桥

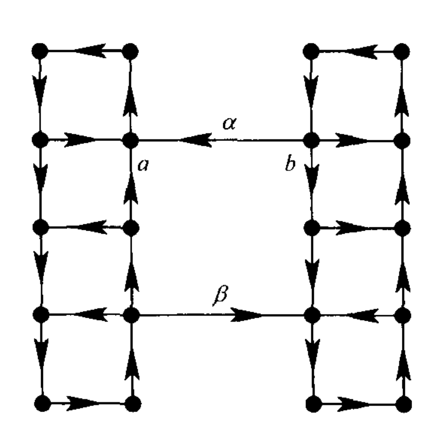


图 11.10　图 11.9 中图的合适的单行线分配

现在，假设 G 有下面的性质：它是一个连通图且没有桥. 这些性质能否保证 G 有强连通方向吗? 可以证明，答案是肯定的，我们概括为如下定理.

定理 11.1(Robbins[1993])　*图 G 有强连通方向，当且仅当 G 是连通的且没有桥.*

我们省略定理 11.1 的证明. 对于其两个不同的证明，可以参见 Roberts[1976, 1978]、Boesch 和 Tindel[1980]. 对于其中一个证明的梗概，参见练习 13.

11.2.2　深度优先搜索算法

在某种意义下，Robbins 定理完全解决了我们提出的问题. 然而，其本身不是非常有用的结果. 因为这个定理只是说明对于一个连通的无桥图，存在单行线分配，但是它没有告知我们如何寻找这样的分配. 本节给出在存在分配时，寻找这样的分配的高效算法.

假设 $G=(V, E)$是任意图. 我们将描述一个定向 G 的过程，这个过程可以完成，当且仅当

G是连通的．如果G没有桥，所产生的方向是强连通的．关于第二个结论的证明，参见 Roberts [1976]．这个过程使用11.1.1节所给出的深度优先搜索过程标签顶点并标记边．如果标签过程不能完成，那么G是非连通的，因此，没有强连通方向．下一步是利用标签和标记边的集合T定义一个方向．假设一条边是使用标签i和j标记的．如果这条边在T中，即它被标记，那么我们定向它为从数字小到数字大，即如果i小于j则从i到j，否则从j到i．如果这条边不在T中，即它没有被标记，那么我们定向它为从较大的数字到较小的数字．利用图11.1的标签和标记来说明这一过程．图11.11给出所对应的方向．注意我们定向从a到b，因为边$\{a, b\}$在T中，而且a得到标签1，这个数字小于b得到的标签2．然而，我们定向从d到b，因为边$\{d, b\}$不在T中，而且d的标签是5，即5大于b的标签2.

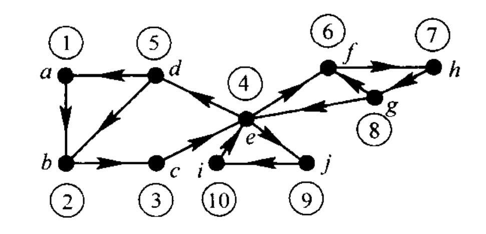


图 11.11　通过标签和标记得到的图11.1中的图G的强连通方向

如果这一标签过程可以完成，即G是连通的，那么这一算法总可以给图G指定一个方向．然而，只有在G没有桥时，才保证产生一个强连通方向．如果假设给定一个标签和T中边的记录，定向边的决策所花费的时间总量等于深度优先搜索过程中用于标签和记录T中的边的每一步所花费的时间，那么，对于有n个顶点和e条边的图，这一算法的计算复杂度$g(n, e)$至多是深度优先搜索过程的计算复杂度加上图G中的边的数量．在11.1.2节中，我们证明了$f(n, e)$至多是$n+2e$．因此，$g(n, e)$至多是$n+3e$．［这又是一个$O(n, e)$算法.］仅用n来计算，这个复杂度至多等于$3/2n^2-3/2n+n$，因为$e\leqslant\binom{n}{2}$．因此，我们有一个多项式界的算法．［它是$O(n^2)$.］对于其他算法，参见 Boesch 和 Tindell[1980]、Chung，Garey 和 Tarjan[1985]．关于强连通方向的平行算法使用的信息，可以参见 Atallah[1984]和 Vishkin[1985]，还可以参考 Karp 和 Ramachandran[1990]中的总结.

619～620

11.2.3　高效单行线分配

并不是每一种城市街道的单行线分配都是高效的．例如，考虑如图11.10中的图所示的单行线分配．这一单行线分配可以通过我们描述的算法(练习4)而得到．如果在位置a处的人想要到达位置b，他曾经可以很快地做到这一点，但是，现在，必须得走很长一段路．简言之，这种分配满足我们建立的标准，可以从任意点到达任意另一个点，然而对于交通流量问题它没有给出一个高效解．一般地，高效单行线分配是这样的分配，在这一分配中，“总体上”经过的距离不会太长.

存在很多关于高效性的可能概念．我们将使这一概念更精确．这些概念都是建立在距离的基础上的．如果G是一个图，回想一下，顶点u和v的距离$d(u, v)$是在G中u和v之间的最短链的长度(边的数量)．如果D是有向图，那么从u到v的距离$\overline{d}(u, v)$是在D中从u到v的最短路径的长度．我们要研究的唯一定义高效性的方法是说，一个方向是最高效的，如果它最小化某个以距离定义的“目标函数”．表11.2给出若干这样的目标函数的例子．最小化其中一些目标函数的问题是等价的．例如，最小化(2)等价于最小化(3).

可以证明，我们处于“坏消息-更坏消息”的情况．坏信息就是，根据表11.2中的任意一个标准还没有找到最高效强连通方向的好算法．更坏的信息是其中某些优化问题是真正的难题．例如，使用2.18节的术语，Chvatal 和 Thomassen[1978]指出寻找最小化(1)的强连通方向是NP难题.

表 11.2　若干强连通方向的目标函数

目标函数	说　明
(1) $\max\limits_{u,v} \bar{d}(u,v)$	经过的最大距离
(2) $\sum\limits_{u,v} \bar{d}(u, v)$	经过距离之和
(3) $\frac{1}{n^2-n}\sum\limits_{u,v} \bar{d}(u, v)$	经过的平均距离
(4) $\max\limits_{u,v}[\bar{d}(u,v)-d(u,v)]$	最大距离差
(5) $\sum\limits_{u,v} [\bar{d}(u, v)-d(u, v)]$	距离差的和
(6) $\frac{1}{n^2-n}\sum\limits_{u,v} [\bar{d}(u, v)-d(u, v)]$	平均距离差
(7) $\sum\limits_{u} \max\limits_{x} \bar{d}(u, x)$	要经过的最大距离之和
(8) $\sum\limits_{u} [\max\limits_{x} \bar{d}(u, x)-\max\limits_{x} d(u, x)]$	要经过的最大距离差的和

还应该强调的是，高效性也不总是单行线分配的目标. 在全美国国家公园系统中，交通阻塞是一个非常严重的问题. 美国国家公园服务局实施的解决方法是，尝试着鼓励人们不要驱车前往国家公园参观. 通过设计一个效率非常低的单行线分配可以做到这一点，这一分配使得很难驱车从一个地方到达另一个地方，而且还鼓励人们选择自行车或巴士作为交通工具. 图 11.12给出 1973 年夏天约塞米蒂谷国家公园约塞米段内制定的大致的单行线分配.（另外，为了使道路成为单行线，公园服务局对汽车完全封锁了其他道路. 注意，存在双向道路；使每一条道路都成为单行线的想法显然过于简单. 在这种情况下，同样没有令人满意的分配，使得每一条道路都是单行线. 为什么?)这是一个极低效的系统. 例如，为了从约塞米地小屋到约塞米地村，这一段步行不到 1 英里的距离，你必须驱车经由波宏诺桥的所有街道，这一段路程超过了 8 英里. 然而，却允许公园的巴士从约塞米地小屋直接通往约塞米地村，因而很多人改乘巴士.

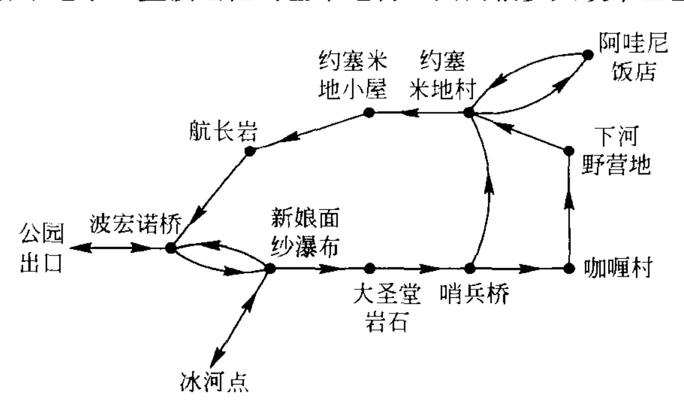


图 11.12　1973 年夏天，约塞米蒂谷汽车交通的单行线分配(注意：巴士可以在约塞米地小屋和约塞米地村之间，以及约塞米地村和哨兵桥之间的道路上双向行驶)

从数学的角度看，寻找低效(最低效)的单行线分配的算法是有益的，如同寻找高效(最高效)分配一样. 遗憾的是，我们同样不知道这样的低效分配算法.

11.2.4 栅格的高效单行线分配

如果一个问题被证明是难题，那么我们可以研究这一问题的更简单的版本，在特殊情况下解决这一问题等等. 这就提出这样的问题：如果我们知道一个图有特殊的结构，那么是否有寻找最高效的强连通方向问题的好的求解方法呢？当然有意义的结构一定与有意义的应用有关. 需要探讨的最普通的结构是对应于有南北街和东西街的城市的栅格图. 我们接下来探讨这些图. 为了固定一个标记法，我们设 $G_{n1,n2}$ 是由 n_1+1 条东西街和 n_2+1 条南北街组成的图. 因此，$G_{3,5}$ 是一个4×6栅格(这一略显笨拙的标记法借用于此课题的有关文献，它在某些证明中很有用).

我们总可以使用蛮力来寻找到最优的强连通方向，即尝试所有方向. 然而，蛮力算法可能非常不现实. 对于有 n 条东西街和 n 条南北街组成的栅格图 $G_{n-1,n-1}$，考虑尝试所有方向的蛮力算法. 需要对每一个这样的方向检测强连通性，如果是强连通的，那么计算最小化的目标函数. 这样的栅格图有多少方向呢？

为了计数栅格图 $G_{n-1,n-1}$ 的方向数量，我们观察到，在每一条东西街上有 $n-1$ 条边，每一条南北街上有 $n-1$ 条边. 因此，边 e 的数量由下式给出：

$$e = 2n(n-1) = 2n^2 - 2n$$

(参见练习 15). 计算出边数之后，我们注意到，$G_{n-1,n-1}$ 的方向数量由乘法规则给出. 因为每一条边有两种可能的方向，总的方向数量是

621～623

$$\overbrace{2\times 2\times\cdots\times 2}^{e\text{次}} = 2^e = 2^{2n^2-2n}.$$

即使我们每秒钟能够检测 10 亿个方向，那么当 n 仅为 7 时，一个简单的计算表明这也要花上 500 000 000 多年才能检测所有方向. 尝试所有方向的蛮力方法显然是不现实的. 即使是计算的速度发生巨大的改变，也无法改变这一结论(作为旁注，我们可以得出结论：在执行一个算法之前，我们总是应该计数这一算法花费的步骤数量).

对于表 11.2 中定义的每一个度量或目标函数，在最小化这一度量的意义下，非常具体的问题就是寻找栅格 $G_{n1,n2}$ 的最优方向. 令人惊讶的是，回答这一问题花费了很长的时间，而且只对其中一些度量找到了答案. 寻找有 2 条东西街和 $n+1$ 条南北街的栅格 $G_{1,n}$ 的强连通方向的第一次尝试开始于 1973 年. 而一直到很长一段时间以后，Roberts 和 Xu[1988，1989，1992，1994] 和 Han[1989]发表了一系列文章，才给出了这些问题的一些解，至少给出表 11.2 中的度量(1)和(8)的一些解.

这里，对于奇数 n，描述 $G_{1,n}$ 情况的最优解非常有益. 可以证明，对于这种情况，最小化(1)或(8)的任意解都在这个栅格的外侧有一个循环(练习 18). 因此，剩下的问题就是决定内部南北街的方向. 我们的直觉是，使这些内部的街道的方向交错. 可以证明，这一交错方向不是最优的. 相反，将中间的两条街道的方向，定为与距离它们最近的(东或西)边界方向相反，而将所有其余内部街道的方向，定为与距离它们最近的边界方向一致是最优的. (参见图 11.13)Roberts 和 Xu[1989]证明，如果 n 是大于等于 5 的奇数，在 $G_{1,n}$ 的所有强连通方向中，这一方向最小化(1)和(8)的度量. 他们还证明，对度量(8)，相对于边界循环的方向，这个方向是唯一的最优方向.

将最优解与交错定向进行比较是有益的，即交错南北街的方向. 例如，当使用度量(1)且

624

$n=5$ 时，最优解得(度量)分为 7，而交错定向的得分为 9. 当 $n=7$ 时，这两个得分分别上升为 9 和 11. 当使用度量(8)时，得分的差异就更大. 当 $n=5$ 时，最优解的得分是 10，而交错定向的

得分是 26；当 $n=7$ 时，得分分别为 12 和 32(容易验证这些得分).

对于其他栅格，Roberts 和 Xu[1988，1989，1992，1994]以及 Han[1989]描述了对于度量(1)和(8)的类似结果. 我们把其中的一些工作编入到练习中. 然而，在其他测量下寻找栅格的最优解仍然是未决问题. 我们很快要进入数学研究的前沿.

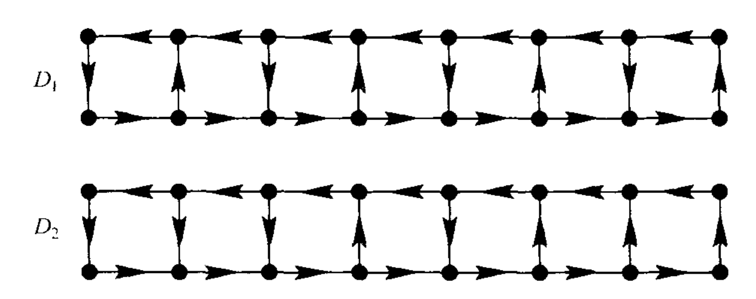


图 11.13　栅格 $G_{1,7}$ 的交错定向 D_1 与最小化度量(1)和(8)的问题的最优解 D_2

研究对东西街和南北街交错定向的自然"纽约市"解是有趣的. 尽管对于度量(1)和(8)纽约市的解决方案不是最优的，在某些其他度量下，纽约市的解决方案比在度量(1)和(8)下的最优解更好. 例如，在度量(3)下，栅格 $G_{5,7}$ 就是如此. 因此，我们还需要做大量的研究.

研究其他图的最优强连通方向也很有意义. 在下一小节里，我们考虑环状城市，即有环形公路及从城市的中心向外延伸的放射状线路的城市.

我们的简化数学模型省略了很多复杂的层次. 例如，我们的模型不考虑不同街道或不同距离的交通量，或穿行不同街道时所需的时间. 考虑另外一种复杂性，Roberts 和 Xu[1994]所描述的一些最优解，包含弧(x, y)和(z, y). 想象在 y 这样的交叉路口处的问题. 如果我们对这样的交叉路口增加若干障碍，那么我们得到不同的优化问题.

11.2.5　环状城市和互联网中的通信

寻找最高效的单行线分配问题还用于研究环状城市，即有环形公路及从城市中心向外延伸的放射状线路的城市. 特别地，我们将讨论图 $AN(c, s)$，这个图围绕着中心有 c 个同心圆，还有穿越所有同心圆的 s 条放射状直线. 图 11.14 给出 $AN(3, 8)$. 我们在练习(练习 23 和练习 24)中给出 $AN(c, s)$的一些结果. 这里，我们只给出一个结果. Bermond 等[2000]对环状城市研究了最小化表 11.2 中的度量(8)的方向. 特别地，他们指出，如果对于某个 k 有 $s=4k$ 且 $c>k+2$，那么存在强连通方向，对于这个方向，$\overline{D}=\max\limits_{u,v}\overline{d}(u, v)=\max\limits_{u,v}d(u, v)=D$，即经过的最大距离不超过定向前所经过的最大距离.

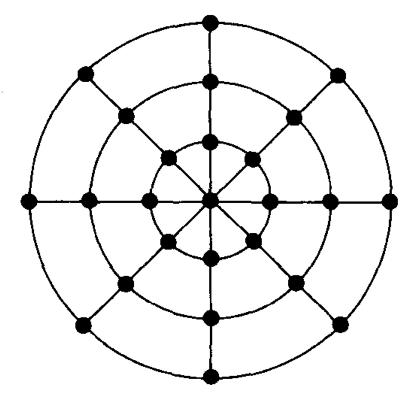


图 11.14　$AN(3, 8)$

识别有 $\overline{D}=D$ 或 $\overline{D}$ 接近 D 的强连通方向的这些图的问题也出现在互联网的通信中. 在"总交换量"里，在这样的一个网络中，最初每一个处理器都有一项信息，这项信息必须分配给其他每一个处理器. 这出现于各种并行处理应用中. 这一过程是通过各处理器之间的同步呼叫序列而实现的. 在每一次呼叫过程中，一个处理器可以与它的所有邻居联络. 在全双工△端口(full-duplex △-port)中，通信可以在邻居之间双向传输. 因此，所有处理器得到所有信息所需的最小步骤数等于 $\max\limits_{u,v}d(u, v)$. 在半双工△端口(half-duplex △-port)中，不允许在给定的链上同时交流. 于是，给定的协议对应于边的一个方向. 在每一步，一个处理器所知道的所有信息，流经有箭头方向的所有处理器. 因此，$\overline{D}$ 等于所有处理器得

625

到所有信息所需的最少步骤数的上界．因为 D 是下界，存在强连通方向，且 $\overline{D}$ 接近 D 的任意网络 G 有一个接近最优性能的协议．

度量(1)或 $\overline{D}$ 还用于其他图的研究．例如，Gutin[1994]、Koh 和 Tan[1996a，b]、Plesnik[1986]发现了所谓的完全多部分图的最优方向，而 McCanna[1988]发现了 n 维立方体的最优方向，这是计算机网络中的一个重要的体系机构．

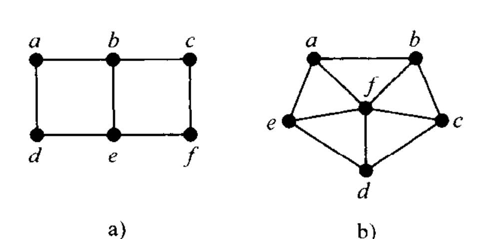
本节练习

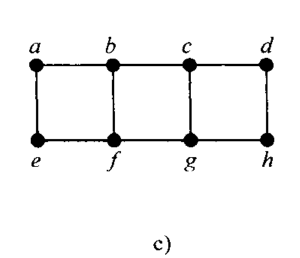
1. 对于图 11.3 中的每一个图，寻找所有桥．
2. 运用本节所描述的过程，寻找图 11.3 中各图的单行线分配．对于每一种情况，验证这一分配是否是强连通的．
3. 对于图 3.23 中的图，重复练习 2．
4. 使用文中所描述的算法，证明图 11.10 所示的单行线分配是可以得到的．
5. 确定一个图有弱连通方向的条件(3.2 节练习 10)． 626
6. 如果一个图有弱连通方向，那么它有多少种弱连通方向．
7. 给出一个有弱连通方向，但不是单连通方向的图的例子(3.2 节练习 9)．
8. 设 G 是一个连通图．G 的*割点*(cut vertex)是有下面性质的顶点：如果你除去这个顶点以及它所属的所有边，那么结果是一个非连通图．对于图 11.3 中的每一个图，寻找所有割点．
9. 带有割点(练习 8)的图有强连通方向吗?
10. 证明或驳证：没有割点(练习 8)的连通图有强连通方向．
11. 证明连通图 G 中的一条边 $\{u, v\}$ 是桥，当且仅当 G 中每一条从 u 到 v 的链都包含边 $\{u, v\}$．
12. 如果 u，v 是有向图 G 中的两个顶点，设 $d_D(u, v)=\overline{d}(u, v)$ 是从 u 到 v 的距离，即是 D 中从 u 到 v 的最短路径的长度．如果不存在从 u 到 v 的路径，那么这一距离无定义．类似地，如果 u，v 是图 G 中的两个顶点，那么从 u 到 v 的距离 $d_D(u, v)$ 是 G 中从 u 到 v 的最短链的长度．而且，如果不存在链，那么距离无定义．
   (a)证明 $d_D(u, v)$ 可能与 $d_D(v, u)$ 不同．
   (b)证明 $d_G(u, v)$ 总是等于 $d_G(v, u)$．
   (c)证明如果 v 从 u 而 w 从 v 是可达的，那么 $d_D(u, w)\leqslant d_D(u, v)+d_D(v, w)$．
13. 在这一练习中，我们考虑一开始就有一些街道是单行线的城市的情况．我们的方法基于 Boesch 和 Tindel[1980]．一个*混合图*(mixed graph)是由顶点的集合组成的，其中的一些顶点被单向弧连接，而另外一些是由无方向的边连接(弧对应于单行线，边对应于双行线)．混合图 G 可以被翻译成有向图 $D(G)$，其方法是设每一条边被两条弧取代，每个方向上有一条弧．G 称为是*强连通*(strongly connected)的，当且仅当 $D(G)$ 是强连通的，如果 $D(G)$ 是弱连通的，则 G 是连通的．连通混合图中的一条边 α 称为是桥(bridge)，如果消除 α 但不消除其端点，那么结果是一个非连通的混合图．
   (a)假设 G 是强连通混合图，而且 $\{u, v\}$ 是 G 的一条边．设 D' 是从 $D(G)$ 通过删除弧 (u, v) 和 (v, u) 但不删除顶点 u 和 v 而得到的有向图．设 A 是通过 D' 中的路径从 u 可达的所有不同于 u 的顶点的集合．类似的，B 定义为从 v 可达的所有不同于 v 的顶点的集合．假设 u 不在 B 中而 v 不在 A 中．证明边 $\{u, v\}$ 一定是 G 的桥．
   (b)使用(a)的结果证明下面的 Boesch 和 Tindell[1980]定理：如果 G 是强连通混合图，且 $\{u, v\}$ 是 G 的一条边，但不是桥，那么存在 $\{u, v\}$ 的方向使得产生的混合图仍然是强连通的．
   (c)根据(b)证明每一个没有桥的强连通图有强连通方向．
   (d)把(c)的证明翻译成寻找连通无桥图的强连通方向的算法． 627
   (e)对于图 11.3 中的图，说明(d)的算法．
   (f)(d)中算法的复杂度是什么? 这一算法与文中描述的算法相比较如何?
14. 确定下面各图的方向数量(不一定是强连通的)：

(a)$G_{2,n}$　　　　(b)$G_{5,11}$　　　　(c)$AN(4, 6)$

15. 使用定理 3.1 计算栅格图 $G_{n-1,n-1}$ 的边数.

16. 对于表 11.2 中的每一个度量(1)～(8)，确定图 11.15 中的每一个图的最高效强连通方向.





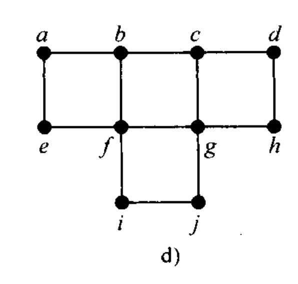


图 11.15　11.2 节练习的图

17. 对于图 11.16 的方向，计算表 11.2 的度量(1)～(8). 这些图分别是类似于图 11.13 中 $G_{1,7}$ 的 D_1 和 D_2 的交错方向和最优解.

18. 证明对于 $G_{1,n}$，n 是奇数的情况，最小化表 11.2 中的度量(1)或(8)的任意解在这个栅格的外侧有一个循环.

19. 当 n 是偶数时，设 $G_{1,n}$ 的方向 O_1 是按如下方法得到的. 使用逆时针循环定向栅格的外侧. 设 $k+1$ 是 $(n/2)+1$. 定向第$(k-1)$条南北街道的方向为向上，第$(k+2)$条南北街道的方向为向下. 所有其他街道定向与第一条或第$(n+1)$条南北街中靠近的一条方向一致. Roberts 和 Xu[1989]证明对于 $n \geqslant 10$，O_1 对表 11.2 中的度量(1)和(8)是最优的.

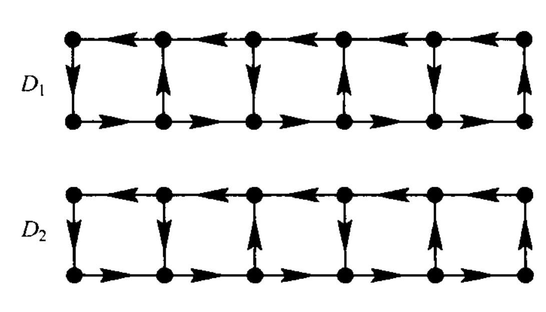


图 11.16　栅格 $G_{1,5}$ 的方向

(a)画出 $G_{1,10}$ 的 O_1.

(b)对于 O_1，计算表 11.2 中的度量(1)和(8).

628

(c)为什么交错的南北街，连同外面的逆时针循环方向，不是 $G_{1,n}$(n 为偶数)的这些度量的最优解?

20. 当 n 是奇数时，设 $G_{2,n}$ 的方向 O_2 是按如下方法得到的. 定向顶部和底部的东西街为从右向左，而中间的东西街为从左向右. 如下定向连接顶部和中间东西街道的南北街：设 $k=(n/2)-1$. 街道 1 向下，街道$n+1$向上，“中间街道”k 和 $k+1$ 的方向不同于街道 1 和街道 $n+1$ 中的较近者(有相反的方向)，而所有其他街道与街道 1 和街道 $n+1$ 中较近者有相同方向. 如下定向连接中间街道和底部东西街道的南北街道：街道 1 向上，街道 $n+1$ 向下，街道 $k-1$ 和 $k+1$ 与街道 1 和街道 $n+1$ 的较近者相反，而所有其余街道的方向与街道 1 和街道 $n+1$ 中较近者有相同方向. Roberts 和 Xu[1989]证明对于 $n \geqslant 7$，O_2 是表 11.2 中的度量(1)和(8)下的最优方向.

(a)画出 $G_{2,7}$ 的 O_2.　　　　(b)画出 $G_{2,9}$ 的 O_2.

(c)对于 $G_{2,7}$，计算表 11.2 中的度量(1)和(8).

(d)对于 $G_{2,9}$，计算表 11.2 中的度量(1)和(8).

(e)假设你交错东西街的方向，使顶部和底部的街道从右向左，中间的街道从左到右；并且交错南北街的方向，街道 1 一直向下，街道 2 一直向上，以此类推. 为什么这不是一个合适的定向?

(f)通过反向定向底部南北街道 1 和南北街道 $n+1$ 的一半来修改(e)中的“交错”方向. 在这一修改的交错方向和 O_2 下，比较 $G_{2,7}$ 的表 11.2 中的度量(1)和(8).

(g)在 O_2 下，我们曾寻找到有弧(x, y)和(z, y)的方向吗?

(h)对于 O_2 及 $G_{2,7}$ 和(f)的修改交错方向，计算表 11.2 中的度量(2)～(7). 哪个更好些?

21. 设 $\bar{d}(u)=\max\limits_{x} \bar{d}(u, x)$，$d(u)=\max\limits_{x} d(u, x)$，$m(u)=\bar{d}(u)-d(u)$. 这时，表 11.2 中的度量(8) 是 $\sum\limits_{u} m(u)$.

629 $G_{n1,n2}$中的隅角点(corner point)是连接东西街和南北街的交叉路口处的点之一. 图 G 的方向 O 内从 u 到 v 的一条极限路径(limiting path)是长度$\overline{d}(u, v)=d(u, v)$的路径.

(a)设 w 是 $G_{n1,n2}$ 中离 u 最远的隅角点，而且在 O 内不存在从 u 到 w 的极限路径，证明$m(u)\geq 2$.

(b)利用练习 20 中的方向 O_2 说明(a)的结果.

(c)对于练习 20 中的(f)的方向，重复(b).

22. 假设 $n_1=2k_1+1$ 是奇数，$i=k_1$ 或 k_1+1. 假设 w 是 $G_{n1,n2}$ 中距位于第 i 条东西街和第 j 条南北街上的点 u 最远的隅角点(练习 21)，而 w'是在相同的南北街上与 w 相对的隅角点.

(a)证明当不存在从 u 到 w'的极限路径时，$m(u)\geq 1$(参见练习 21).

(b)利用练习 20 中的方向 O_2 说明(a)的结果.

(c)对于练习 20 中的(f)的方向，重复(b).

23. (Bermond 等[2000])设 $D=\max_{u,v} d(u, v)$且考虑图 $AN(c, s)$.

(a)证明如果 $c\leq\lfloor s/4 \rfloor$，那么 $D=2c$.

(b)证明如果 $c\geq\lfloor s/4 \rfloor$，那么 $D=c+\lfloor s/4 \rfloor$.

24. (Bermond 等[2000])设 O_3 是如下 $AN(c, s)$的方向：第一条辐线的方向为从外圆到中心，其余辐线的方向交错. 如果 s 是奇数，那么辐线 1 和 s 将有相同的方向. 如下定向圆的方向：(从中心开始)第一个圆定向为沿逆时针方向，接下来的圆沿顺时针方向，下面的圆沿逆时针方向，以此类推，一直到倒数第二个圆. 对于外圆，定向弧从辐线 2 到辐线 1，辐线 2 到辐线 3，辐线 4 到辐线 3，辐线 4 到辐线 5，以此类推. 如果 s 是奇数，那么从辐线 2 到辐线 1 再到辐线 s 的外圆部分形成唯一的长度为 2 的路径.

(a)如果对于 O_3，$\overline{D}=\max_{u,v}\overline{d}(u, v)$，对于 $AN(c, s)$，$D=\max_{u,v} d(u, v)$，证明如果 $c\leq k$，则 $\overline{D}\leq D+2$，其中 $s=4k+r$，$r\leq 4$.

(b)通过与 O_3 相同的推理，证明如果 s 是奇数，且 $c=k+1$，则 $\overline{D}\leq D+3$.

25. 对于运输网络、通信网络、电话网络和电网等等，主要关心的一点是构造它们使得它们不容易被破坏. 受到桥的概念的启发，我们在练习 25～33 中，对由于弧或顶点的删除而引发的连通性破坏感兴趣. 我们说有向图 G 处于连通范畴 3(connectedness category 3)，如果它是强连通的，如果它是单连通而不是强连通的，则它处于范畴 2(category 2)，如果它是弱连通而不是单连通的，那么它处于范畴 1(category 1)，如果它不是弱连通的，那么它处于范畴 0(category 0). 确定图 3.7 中的每一个有向图的连通范畴.

26. 如果有向图 D 处于连通范畴 i(练习 25)，我们说弧(u, v)是(i, j)弧((i, j)-arc)，如果删除(u, v)(但是不删除顶点 u 和 v)产生范畴 j 的有向图. 对于每一对(i, j)，$i=0, 1, 2, 3$，$j=0, 1, 2, 3$，或者给出一个有(i, j)弧的有向图的例子，或者证明不存在这样的有向图.

630

27. 如果有向图 D 处于连通范畴 i，我们说顶点 u 是(i, j)顶点((i, j)-vertex)，如果 D 中除 u 外的其他顶点所生成的子图处于范畴 j. 对于每个对(i, j)，$i=0, 1, 2, 3$，$j=0, 1, 2, 3$，或者给出有(i, j)顶点的有向图的例子，或者证明不存在这样的有向图.

28. 我们定义连通范畴 1，2 或 3 的有向图的弧易损度(arc vulnerability)为满足下面条件的弧的最小数量，这些弧的删除导致更低连通范畴的有向图. 对于图 11.17 中的每一个有向图，确定它的弧易损度.

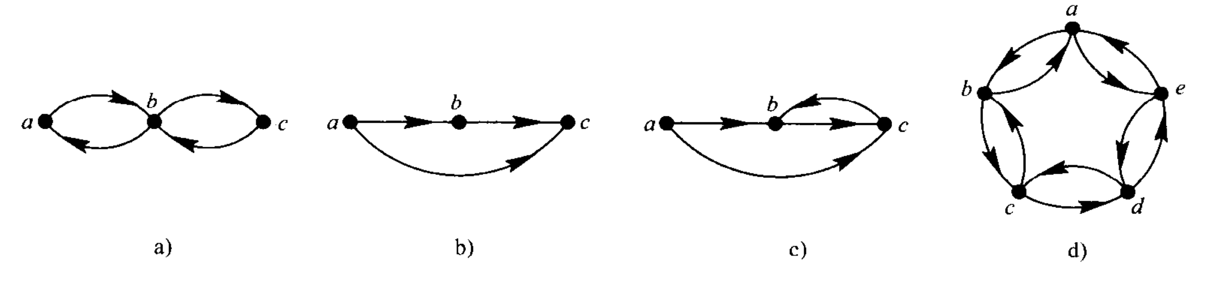


图 11.17 11.2 节练习的有向图

29. 给出一个弧易损度为 4 的有向图的例子.
30. 对于每一个 k，给出弧易损度为 k 的有向图的例子.
31. 给出一个顶点数量为 n，弧易损度为 $n-1$ 的有向图 D 的例子. 存在顶点个数为 n 且弧易损度为 n 的强连通有向图 D 吗?
32. (a)如果 D 是强连通的，D 的弧易损度与顶点的最小**入度数**(indegree)之间的关系是什么? 顶点的最小入度数就是顶点的进入弧的最小数量.
    (b)证明强连通有向图的弧易损度至多是 a/n，其中 a 是弧的数量，n 是顶点的数量.
33. (Whitney[1932])如果存在有向图 D 的顶点集合，其中的顶点的删除导致连通范畴低于 D 的有向图，那么我们定义 D 的**顶点易损度**(vertex vulnerability)为这样的顶点集合中最小的集合的大小. 否则，顶点易损度无定义.
    (a)证明如果 D 是弱连通的，且至少存在一对 $u\neq v$，使得(u, v)和(v, u)都不是 D 的弧，那么 D 的顶点易损度小于或等于 D 中任意顶点的最小总度数[顶点 x 的总度数是 x 的入度数(进入弧的数量)和出度数(离开弧的数量)的和].
    (b)证明顶点易损度和最小总度数可以是不同的.
    (c)为什么我们需要假设至少存在一对 $u\neq v$，使得(u, v)和(v, u)都不是 D 的弧?

631

11.3　欧拉链和欧拉路径

11.3.1　哥尼斯堡桥问题

图论是著名数学家欧拉[1736]在解决著名的哥尼斯堡桥问题[⊖]时创建的. 欧拉的技术已应用于街道清扫、计算机机械制图、RNA 链、编码、电信和其他现代学科的研究中，我们将在 11.4 节中概述其中的一些应用. 在这里，我们给出哥尼斯堡桥问题，然后给出由它的解所引出的通用技术.

如图 11.18 所示，哥尼斯堡城有 7 座桥梁，将普里高里河中的岛屿和岸边连接起来. 居民们想要知道是否有可能从某一点出发，正好徒步穿越每一座桥梁一次，最后返回到出发地点. 欧拉把这一问题翻译成一个图论问题，其方法是设各个陆地区域是顶点，通过连接两个陆地区域的每座桥为一条边连接两个顶点. 如图 11.19 所示的结果对象称为**多重图**(multigraph). 当两个顶点 x 和 y 之间或从顶点 x 到顶点 y 之间允许存在多条边或多条弧时，我们将使用术语多重图或**多重有向图**(multidigraph). 多重图或多重有向图的连通性的概念，诸如链、回路、路径、循环、分支等等都如图或有向图的定义(然而，请注意，在多重图中，如果在 a 和 b 之间存在两条边，我们有一条从 a 到 b 再到 a 的回路). 这里，我们允许存在环.

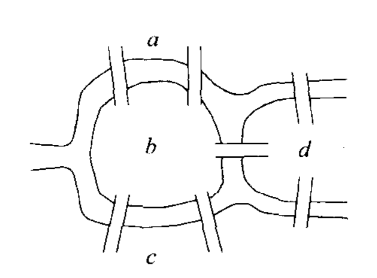


图 11.18　哥尼斯堡桥

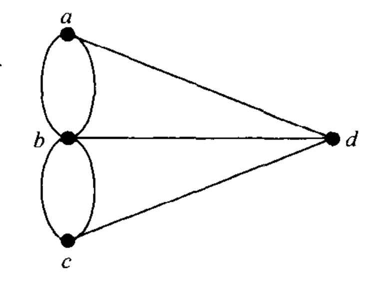


图 11.19　得自于图 11.18 的多重图

我们说多重图 G 中的一条**链**(chain)或多重有向图 D 中的一条**路径**(path)是**欧拉的**(Eulerian)，如果它使用 G 的每一条边或 D 的每一条弧一次且仅一次. 欧拉观察到哥尼斯堡市民

632

⊖ 参见"The Königsberg Bridges", *Scientific American*, 189(1953), 66～70.

是在图 11.9 所示的多重图中寻找一条欧拉闭链(eulerian closed chain). 他提出了下面的一般问题：什么时候一个多重图 G 有一条欧拉闭链？我们从尝试回答这个问题开始. 显然，如果 G 有欧拉闭链，那么 G 一定是相对于孤立顶点连通的(connected up to isolated vertices)[㊀]，即至多有一个 G 的分支有边. 另外，一条欧拉闭链，离开每一个顶点 x 的频率必须等于它进入 x 的频率，因此每一个顶点 x 必须有偶数度，读者将回想到，其中 x 的度是 x 的邻居的数量(除从 x 到 x 环之外，对此我们把 x 的度加 2). 欧拉发现下面的定理.

定理 11.2(欧拉)[㊁] *多重图 G 有欧拉闭链，当且仅当 G 是相对于孤立顶点连通的，而且 G 的每一个顶点都有偶数度.*

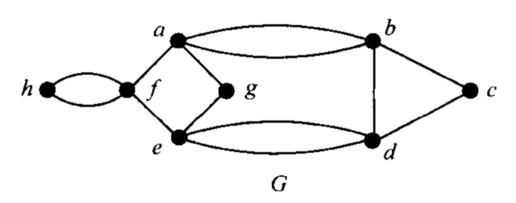


图 11.20 有欧拉闭链的多重图

在证明定理 11.2 的充分条件之前，我们观察到图 11.19 的多重图没有闭欧拉链. 因为顶点 a 有奇数度. 因此，哥尼斯堡市民不可能完成他们的行走. 为了进一步说明这一定理，我们注意到图 11.20 给出的多重图 G 是连通的，且每一个顶点有偶数度. 一条欧拉闭链是 a，b，c，d，e，f，h，f，a，g，e，d，b，a. 如果多重多图 G 的每一个顶点有偶数度，那么我们说这个 G 是偶的(even).

11.3.2 寻找欧拉闭链的算法

证明定理 11.2 的充分性的一个好方法是描述在偶多重图中寻找欧拉闭链的过程. 从有邻居的任意顶点 x 开始，并选择连接 x 的任意一条边，比如说是$\{x, y\}$. 接下来，选择连接 y 的任意一条边$\{y, z\}$，此时要保证不选取前面已使用过的边. 从顶点 z 继续下去，选取前面没有使用且连接这个顶点的边. 现在我们每经过一个顶点，我们就用去两条邻接的边. 因此，连接 x 之外的任意顶点的未使用的边的数量总是剩下偶数个. 因此，每当我们进入一个 x 之外的顶点，我们都可以从它出去. 因为我们从 x 开始，所以连接 x 的未使用边的数量总是保持奇数. 继续这个过程直到它不可能继续下去. 现在，因为我们每一次进入任意一个 x 之外的顶点，我们都能出去. 这个过程结束的唯一方法是返回到 x. 让我们在图 11.20 中的多重图 G 上说明它. 假设 x 是 c. 我们可以使用边$\{c, b\}$到达 b，然后使用边$\{b, a\}$到达 a，再使用边$\{a, g\}$到达 g，使用边$\{g, e\}$到达 e，使用边$\{e, d\}$到达 d，使用边$\{d, c\}$返回 c. 此时，我们无法继续前进. 注意，我们已经发现了一个闭链 c，b，a，g，e，d，c，这条链开始并结束于 c，而且这条链使用 G 的每一条边至多一次. 然而，这条链没有用尽 G 的所有边. 我们称这一过程为"寻找从 x 至 x 的闭链"，或简称为 CL CHAIN(x, x).

633

现在，这一算法可以继续. 注意在此阶段，我们有一条从 x 到 x 的链 C. 另外，每一个顶点都有偶数条没有使用过的连接边. 因为这个图是相对于孤立顶点连通的，如果存在任意未使用边，那么一定至少存在一条未使用边连接链 C 上的一个顶点 u. 在未使用边的多重图内，我们运用过程 CL CHAIN(u, u)去寻找从 u 到 u 的闭链 D，这条链使用前面未使用的边至多一次. 现在我们可以修改原来的从 x 到 x 的闭链 C，其方法是当我们首次碰到 u 时，插入"便道"D. 我们得到一条新的从 x 到 x 的闭链 C'，它使用 G 的每一条边至多一次. 如果还存在没有使用的边，那么我们可以重复上述过程. 最终我们一定用尽原来多重图的所有边. 继续我们的例子，注意到至

㊀ 如果一个顶点没有邻居，那么它被称为孤立顶点.

㊁ 注意，欧拉闭链不一定通过图中每一个顶点. 例如，在图 11.22 中加入孤立顶点 i 及孤立顶点 j 后，a，b，c，d，e，f，h，f，a，g，e，d，b，a 仍是一条欧拉闭链. 这就是本书使用"相对于孤立顶点连通"这一不常见术语的原因. 另外，本定理的陈述比其他离散数学和组合数学的相应定理强. ——译者注

此我们已得到闭链 $C=c, b, a, g, e, d, c$. 连接这条链的一条未使用边是边$\{a, f\}$. 因为 a 在 C 上，所以我们寻找一条从 a 到 a 的未使用边的闭链. 可以利用我们前面的过程 CL CHAIN (a, a)寻找这样的链. 一个例子就是 $D=a, f, e, d, b, a$. 注意，这里，我们使用的是$\{e, d\}$和$\{b, a\}$的第二条边. 前面的边已使用过了. 我们把便道 D 插入到 C 中，得到闭链 $C'=c, b, a, f, e, d, b, a, g, e, d, c$. 因为仍然存在未使用边，我们再次重复这一过程. 我们在 C'上寻找连接未使用的边的顶点 f. 而且我们使用 CL CHAIN(f, f)寻找一条从 f 到 f 的闭链 f, h, f. 我们把这一便道插入到 C'中得到 $C''=c, b, a, f, h, f, e, d, b, a, g, e, d, c$，这条链使用每一条边一次而且只有一次.

现在形式化上面所述的算法. 我们有一个称为 CL CHAIN(x, x)的子过程，这个过程叙述如下.

算法 11.3：CL CHAIN(x, x)

输入： 多重图 G, G 的未使用边集合 U，其中每一个顶点都在 U 的偶数条边上出现，以及出现在 U 的某些边上的指定顶点 x.

输出： 使用 U 的每一条边至多一次的从 x 到 x 的闭链.

步骤 1. 设置 $v=x$，且输出 x(这里，v 是最后访问的顶点).

步骤 2. 如果 U 中存在一条边$\{v, y\}$，设置 $v=y$，输出 y，从 U 中删除$\{v, y\}$，并重复这一步骤.

步骤 3. 如果 U 中不存在边$\{v, y\}$，则停止. 按顺序输出，得到一条我们想要的从 x 到 x 的闭链.

使用算法 11.3，现在我们可以概括整个过程.

[634]

算法 11.4：寻找一条欧拉闭链

输入： 相对于孤立顶点连通的偶多重图 G.

输出： 欧拉闭链.

步骤 1. 任选一个有邻居的顶点 x(如果不存在这样的顶点，那么每一个顶点都是孤立的，而且任意一个顶点就定义一条欧拉闭链，输出这一顶点并停止). 设 $U=E(G)$.

步骤 2. 运用 CL CHAIN(x, x)得到一条链 C.

步骤 3. 从 U 中删除 C 中的所有边(从技术上看，这在步骤 2 中已完成了).

步骤 4. 如果 $U=$空集，停止并输出 C. 如果 $U\neq$空集，那么 C 上寻找一个在 U 中有连接边的顶点 u，前进到步骤 5.

步骤 5. 运用 CL CHAIN(u, u)得到一条链 D.

步骤 6. 在第一个被访问点 u 处插入便道 D，重新定义 C，从 U 中删除 D 的所有边，并返回到步骤 4.

对于这一算法的更形式化的描述，请参见 Even[1979]. Even 指出在最坏的情况下，完成这一算法所用的步骤数是常量 k 乘以边数 e. 因此，如果 G 是一个图(即如果没有多重边)，上面的算法的计算复杂度是

$$ke \leqslant k\frac{n(n-1)}{2} = \frac{k}{2}n^2 - \frac{k}{2}n,$$

一个关于 n 的多项式界. [用 2.18 节的术语，这是一个 $O(e)$，或对于图来说是 $O(n^2)$算法.]因为显然这一算法可行，从而证明了定理 11.2.

寻找一条欧拉闭链的另外一个算法要使用 11.1 节的深度优先搜索过程. 我们把这样一个算

法的陈述留给读者(练习 18).

11.3.3　关于欧拉链和欧拉路径的更多结果

下面的定理告诉我们什么时候一个多重图存在一条欧拉链(不一定是封闭的). 我们把其证明留给读者.

定理 11.3(欧拉)　*一个多重多图 G 有一条欧拉链当且仅当如果 G 相对于孤立顶点连通，且奇数度数的顶点的数量等于 0 或 2.*

根据这一定理，图 11.19 的多重图没有欧拉链. 然而，图 11.21 的多重图有欧拉链，因为奇度数的顶点的数量正好为 2.

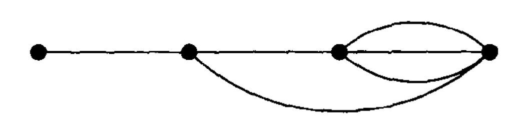


图 11.21　有一条欧拉链的多重图

现在我们对多重有向图陈述类似的结果，并把其证明留给读者. 在这些定理中，一个顶点的入度数(出度数)是进入(出去)的弧的数量. 一个有向图(多重有向图)称为弱连通(weakly connected)，如果忽略弧的所有方向，其结果图(多重图)是连通的.

[635]

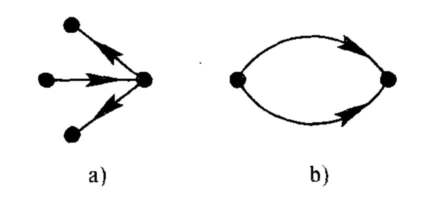


图 11.22　没有欧拉路径的两个多重有向图

定理 11.4(Good[1946])　*一个多重有向图 D 有一条欧拉封闭路径，当且仅当 D 相对于孤立顶点弱连通*[1]*，且对于每一个顶点，入度数等于出度数.*

定理 11.5(Good[1946])　*一个多重有向图 D 有一条欧拉路径，当且仅当 D 相对于孤立顶点弱连通，且除了可能的两个顶点外，对于所有顶点，入度数等于出度数，且至多对于两个顶点，入度数与出度数相差 1.*

通过图 11.22 的两个多重有向图说明定理 11.5. 这两个图都没有欧拉路径. 在第一个例子中，存在 4 个顶点，它们的入度数不同于出度数，在第二个例子中，正好有两个这样的顶点，但是对于每一个这样的顶点，入度数与出度数的差大于 1. 注意定理 11.5 的条件意味着如果对于任意顶点，入度数不等于出度数，那么正好有一个顶点剩余一个入度数，及正好有一个顶点剩余一个出度数(练习 9). 可以证明这两个顶点对应于这条欧拉路径的第一个顶点和最后一个顶点.

注意如果存在环，那么定理 11.2～定理 11.5 成立. 还要注意在一个多重图中，一个环把一个顶点的度数增加 2，而在一个多重有向图中，自循环把入度数增加 1，出度数加 1. 因此，自循环不影响欧拉链(封闭)或路径的存在.

[636]

本节练习

1. 对于图 11.23 所示的每一条有桥的河，构建一个对应的多重图并确定是否有可能从一个给定点出发经过每一座桥一次且仅有一次，并返回到出发点.
2. 图 11.24 的多重图中哪些有一条欧拉闭链? 对于存在欧拉闭链的多重图，寻找一条欧拉闭链.
3. 图 11.24 的多重图中，有的没有欧拉闭链，哪些有一条欧拉链?
4. 图 11.25 的多重有向图中，哪些图有一条欧拉封闭路径? 对于有欧拉封闭路径的多重有向图，寻找一条欧拉封闭路径.

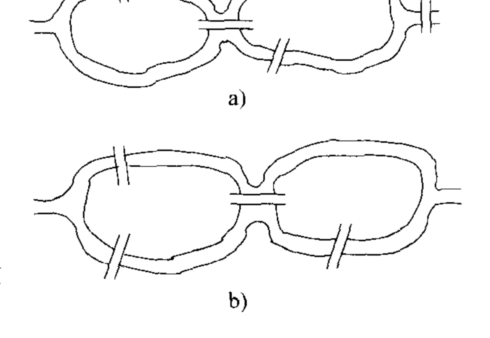


图 11.23　11.3 节练习 1 的河和桥

---

[1] 即当忽略弧的所有方向时，基础多重图是相对于孤立顶点连通的.

5. 图 11.25 的多重有向图中，有的图没有欧拉封闭路径，哪些图有欧拉路径？

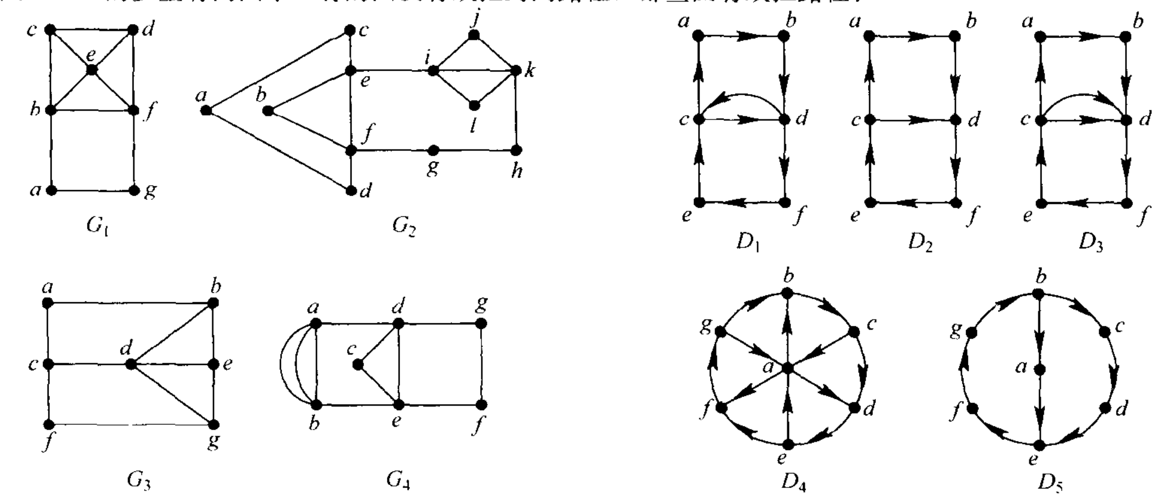


图 11.24 11.3 节练习的多重图

图 11.25 11.3 节练习的多重有向图

6. 对于图 11.26 的每一个多重图 G，对顶点 a 及 $U=E(G)$ 运用子程序 CL CHAIN(a，a).
7. 对于图 11.26 中的每一个多重图，运用定理 11.4 寻找一条欧拉闭链.
8. 你如何修改定理 11.4，以便寻找从 x 到 y 的一条欧拉链.
9. 证明在一个有欧拉路径但没有欧拉封闭路径的多重有向图中，正好有一个顶点剩余一个入度数，正好有一个顶点剩余一个出度数.
10. (a)使你的笔不离开纸面且不重叠，能够画出图 11.27a 吗？
    (b)对于图 11.27b 情况又如何？
    (c)对于图 11.27c 情况又如何？
    (d)对于图 11.27d 情况又如何？

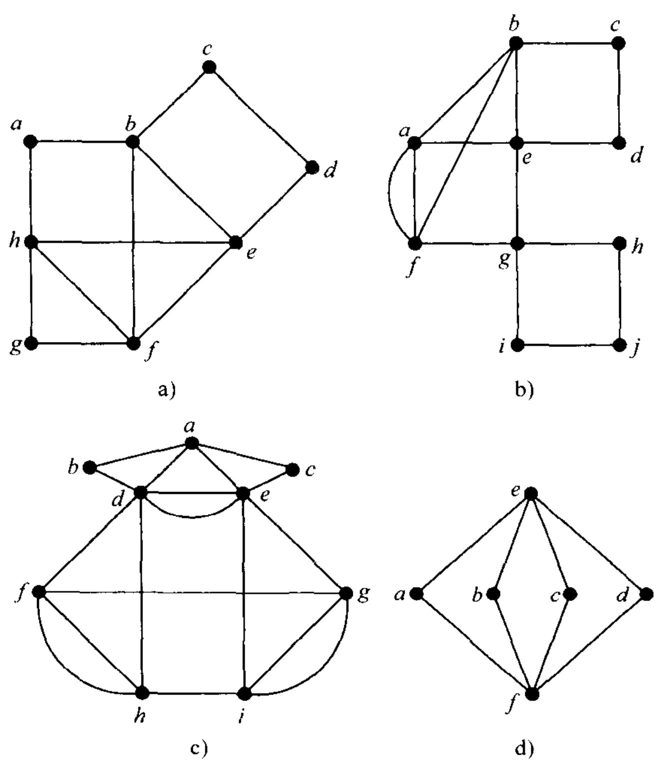


图 11.26 11.3 节练习的多重图

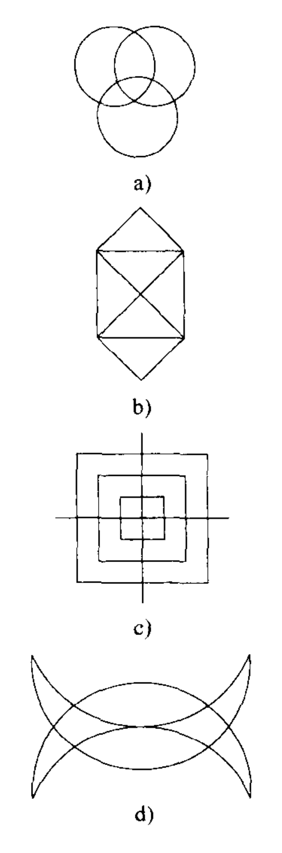


图 11.27 11.3 节练习 10 的画图

637
~
639

11. (Harary 和 Palmer[1973])证明 n 个顶点的标签偶图的数量等于 $n-1$ 个顶点的标签图的数量. [提示：对于一个 $n-1$ 个顶点的标签图，增加标签为 n 的一个顶点，并把它与奇度数的 G 的顶点连接起来.]
12. (a)对于图 11.25 中的有欧拉封闭路径的每一个多重有向图，确定开始和结束于 a 的欧拉封闭路径的数量.
    (b)对于图 11.25 中的没有欧拉封闭路径但是有欧拉路径的每一个多重有向图，确定欧拉路径的数量.
13. 对于图 11.26d 的图，确定开始和结束于 a 的欧拉闭链的数量.
14. 证明没有孤立顶点但有欧拉封闭路径的每一个有向图是强连通的.
15. 证明没有孤立顶点但有欧拉路径的每一个有向图是单连通的.
16. 每一个强连通有向图都有欧拉封闭路径吗？为什么？
17. 每一个单连通有向图都有欧拉路径吗？为什么？
18. 描述使用深度优先搜索过程寻找一条欧拉闭链的算法.
19. 证明定理 11.3 的必要部分.
20. 证明定理 11.3 的充分部分.
21. 证明定理 11.4.
22. 证明定理 11.5.

11.4　欧拉链和欧拉路径的应用

本节，我们给出 11.3 节所陈述概念的各种应用. 本节的各小节是独立的，除 11.4.2 节和 11.4.3 节要依赖于 11.4.1 节之外. 除这之外，各小节的阅读顺序不限. 没有时间研读所有各小节的读者，可以抽出其中的几小节阅读，如 11.4.1 节、11.4.2 节、11.4.4 节和 11.4.6 节.

11.4.1　“中国邮差”问题

从一个邮局出来的邮递员必须把信件送到某个区域的每一个街区，然后返回到这个邮局. 这名邮递员所走过的最短路径是多少？这一问题原本是 Kwan[1962]所研究的，而且传统上一直把这一问题称为“*中国邮差*”*问题*(chinese postman problem)，或只称*邮差问题*(postman problem). 很多投递员、地里播种的农民、道路维修人员等都会遇到类似的问题.

640

我们通过构建每一个顶点表示一个路口、每一条边表示一条街的图 G，来表示邮差问题[⊖]. 为简便起见，假设邮局在路口附近，那么邮递员在这个图中寻找一条闭链，这条闭链开始和结束于对应于邮局附近路口的顶点 x，而且它使用这个图的每一条边至少一次.

如果图 G 有一条欧拉闭链，我们可以选出这条开始于 x 的链，并沿着这条链返回到 x. 没有哪条链能给出更短的路径. 如果不存在这样的欧拉闭链，那么我们可以如下阐述这一问题. 任意一名邮递员的路径都将使用 G 的所有边一次，而且其中某些边可能使用多次. 假设我们按如下方法修改 G：我们可以用在这名邮递员的路径中出现的边的拷贝取代 G 中的每一条边，拷贝次数等于这条边在邮递员的路径出现的次数，或者等价于，给 G 增加足够多的每一条边的拷贝，使得可以形成这名邮递员的完整路径. 那么在得到的多重图中，这名邮递员的路径对应于从 x 到 x 的欧拉闭链. 例如，考虑图 11.28 中的图 G，它代表一个城市的一个四方形区域. 在 G 中没有欧拉闭链，因为顶点 d 的度数为 3. 邮递员的一条可能的路径是闭链 x，h，g，d，e，f，c，b，a，d，e，b，e，h，x，f，x. 这条链对应于图 11.28 所示的第一个多重图(增加的边用虚线表示). 另一条可能的路径是 x，h，e，h，g，d，e，d，g，d，a，b，e，b，c，f，e，f，x. 这条链对应于图 11.28 的第二个多重图. 其中第一条路径较短；等价地，它需要补充较少的 G 的边的拷贝

⊖ 这样做可以简化这一问题. 两侧有房子的 x 和 y 之间的一条街实际上应该被表示成 x 和 y 之间的两条边. 一名邮递员首先沿着这条街的一侧行走，然后再在另一侧向回行走.

(第二条路径通过忽略 d, g, d 部分可变短).

尝试着在 G 中寻找邮递员从 x 到 x 的最短路径的问题，等价于这样的问题：确定加入到 G 中的 G 的边的最小拷贝数量，以便得到一个有欧拉闭链的多重图，即在一个所有顶点都有偶数度的多重图内有一条欧拉闭链. 解决这一组合优化问题的一般方法是，把这个问题翻译成第 12 章将要讨论的类型的最大权匹配问题. 我们将在 12.7.1 节中具体讨论这一翻译. 在我们的例子中，很容易看到，因为存在四个奇数度的顶点，而且没有两个是邻接的，所到至少要加入四条边. 因此，图 11.28的第一个多重图对应于邮递员的最短路径.

在 11.4.3 节中，我们扩展这一问题. 关于"中国邮差"问题讨论的较好文献请参见 Lawler [1976]、Minieka [1978]、Johnson [2000].

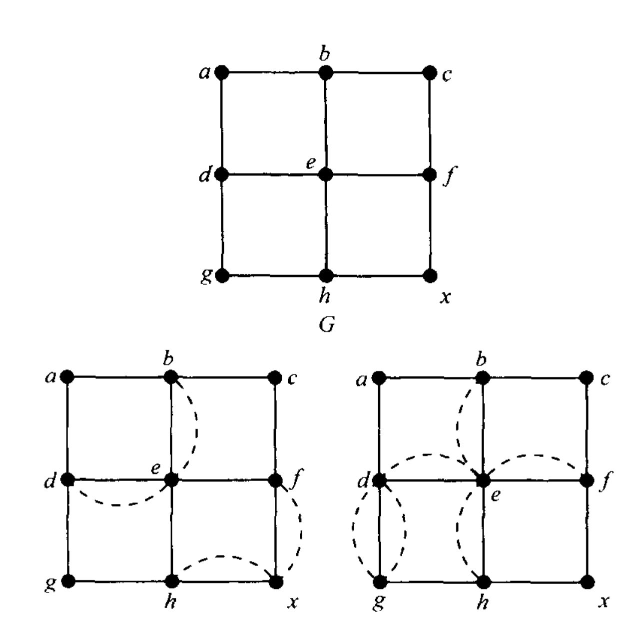


图 11.28　表示一名邮递员所负责的区域图和对应于邮递员路径的两个多重图

11.4.2　计算机绘图

Reingold 和 Tarjan[1981]指出，11.4.1 节所讨论的"中国邮差"问题也会出现在通过计算机⊖对事先指定顶点位置的图的机械绘制中. Reingold 和 Tarjan 所描述的机械绘图包含冲击波的传播问题，在这一问题中，必须画出数以千记的顶点的网格；地图绘制；电网以及活动图表(参见 Reingold 和 Tarjan 的论文作为参考).

当绘图仪随着笔离开纸面移动时，在机械绘图中要浪费大量的时间. 因此，我们要设法最小化从顶点到顶点的笔离开纸面的移动次数. 这正好是下面这样的问题，把边加入到被绘制的图中，以得到带有欧拉闭链的多重图的边数的最小化问题.

11.4.3　街道清扫

组合数学应用的一个较大的领域是城市公共事业的领域. 很多城市每年要花去数以十亿计的美元提供公共服务. 组合数学已应用于很多方面，其中包括火灾和警察局的位置及人员配置，快速运输系统的设计，市政工人班次的安排，为街道清扫和除雪工具安排路线等等. 对于以上各方面的应用例子，可以参见 Beltrami[1977]、Dror[2000]和 Helly[1975]. 我们在例子 3.14 中讨论了回收垃圾的垃圾卡车的路线问题. 这里，我们讨论确定街道清扫和除雪设备的优化路径的问题. 我们参考 Tuucker 和 Bodin[1983]；也可以参见 Liebling[1970].

641 ~ 642

我们具体考虑街道清扫问题. 设在要被清扫的区域的街道路口是多重有向图的顶点. 如果有一条马路从 x 指向 y，那么就包含一条从 x 到 y 的弧. 一般地，单行线给出从 x 到 y 的两条弧，这就是为什么我们得到一个多重有向图. 称这个多重有向图为*马路多重有向图*(curb multidigraph).

在给定的一段时间内，要清扫某些街道. 这些街道所对应的弧定义一个称为*清扫子图*

⊖ 尽管迄今仍用于 CAD 应用，但是笔绘图仪很快变得陈旧，这归功于喷墨和激光打印机的出现.

(sweep suubgraph)的马路多重有向图的子图(根据停止规则，在一个较大的城市诸如纽约城中，要清扫的街道在清扫期间要保持车辆畅通无阻).

现在清扫子图中的任意一条弧与表明清扫对应马路所需要的时间长度的一个数有关. 同样，马路多重有向图中的任意一条弧与表明走过对应马路而不清扫它所需要的时间长度的一个数有关. 我们把这个时间称为空载返回时间(deadheading time).

图 11.29 给出一个马路多重有向图. 其中一些弧是实线表示的，这些弧定义清扫子图. 每一个弧都有在方格中的数字；这个数字是空载返回时间. 实线弧有另外一个在圆圈中的数字，这个数字是清扫时间.

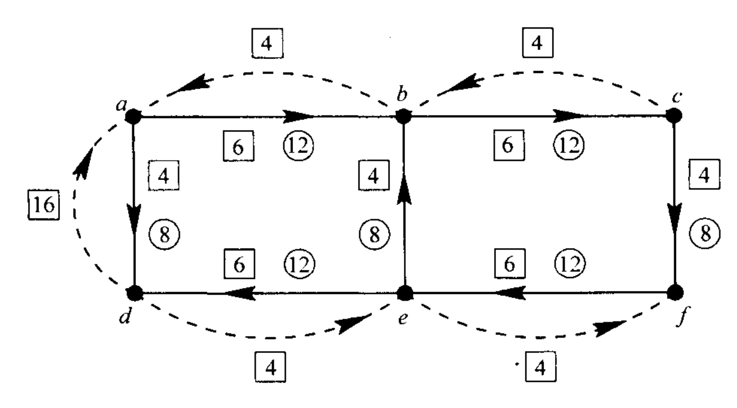


图 11.29 马路多重有向图，实线弧定义清扫子图，方块中的数字表示空载返回时间，而圆圈中的数字表示清扫时间

我们希望寻找一种方法，从某一特定地点(车库)开始，清扫清扫子图中的所有马路，返回到出发地点，并使用尽可能少的时间. 这是 11.4.1 节所讨论的“中国邮差”问题的扩展，而且我们对此的处理方法也是类似的. 我们在马路多重有向子图中，寻找包含清扫子图的所有弧的一条封闭路径. 与任意可接受路径相关的时间是，被清扫弧的清扫时间的总和加上没有被清扫路径中的弧的空载返回时间总和. 注意一条弧可以在这条路径中出现多次. 即使它是实线弧，我们只计数清扫时间一次. 这条弧只在第一次经过时被清扫. 对于这条弧的其他每一次出现，我们计数空载返回时间.

643

如果清扫子图有一条欧拉封闭路径，那么正如在“中国邮差”问题中那样，这条路径必定是最少时间的街道清扫路径. 在我们的例子中，定理 11.4 表明不存在这样的路径. 因为清扫子图中有几个顶点，这几个顶点的入度数不等于出度数. 因此，街道清扫问题的一个解是使用不在清扫子图中的某些弧的马路多重有向图的一条封闭路径 P. 假设为空载，我们把 P 中使用的所有弧加入到清扫子图中，或者是因为它们不在清扫子图中，或是因为它们先前已在 P 中被清扫过. 加入这些弧的次数等于它们所使用的次数. 加入这些弧给出一个多重有向图，在这个多重有向图中，路径 P 是一条欧拉封闭路径. 例如，图 11.29 中的马路多重有向图中的使用了清扫子图的所有弧的一条封闭路径，是由下面的弧给出的：

$$f,e,b,c,f,e,d,a,d,a,b,c,f \tag{11.1}$$

(假设车库是在 f). 路径(11.1)对应于加到清扫子图中的如图 11.30 所示的虚线弧.

一般地，我们希望所加入弧上的空载返回时间总和最小化[⊖]，而且我们想要加入空载返回

⊖ 我们忽略空载返回之外的其他耽搁的源头，例如，与卡车调头相关的耽搁等. 这些耽搁可以通过定义与不同种类的卡车调头相关的罚时体系而引入(参见 Tucker 和 Bodin[1983]).

弧，使得在结果多重有向图中，有一条欧拉封闭路径；即每一个顶点都有入度数等于出度数. Tucker 和 Bodin[1983]给出如何求解其结果的组合优化问题，其方法是把这一问题设为 13.4.1 节所述的运输问题. 这一方法类似于用于 12.7.1 节所述的“中国邮差”问题的方法. 在我们的例子中，一条优化的路线是路径 f, e, d, e, b, a, b, a, d, e, b, c, f. 这一路线所需要的时间总量是清扫的 72 加上空载返回的 20. 路线(11.1)所需要的时间总量是清扫的 72 加上空载返回的 48，这种情况相对不好.

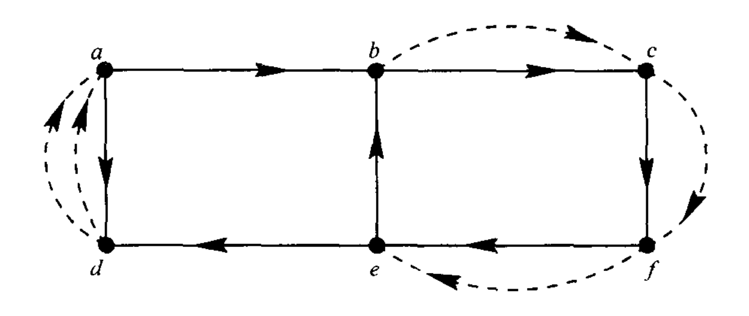


图 11.30　通过增加虚线弧从图 11.29 的清扫子图(对应于路线(11.1))而得的有欧拉封闭路径的多重有向图

644

11.4.4　寻找未知的 RNA/DNA 链

本小节考虑由一个未知串的片段集合，寻找未知的 RNA/DNA 串的问题.

【例 11.1 酶的完全分解】 2.12 节讨论了在运用两个酶之后，给定 RNA 链片段的信息，寻找这个 RNA 链的问题，这两个酶中的一个在每一个 G 链接之后分裂这个链，而另外一个酶在每一个 U 或 C 链接之后分裂这个链. 这里，我们给出如何使用欧拉封闭路径的概念，寻找有给定的 G，U，C 片段的所有 RNA 链.

如果仅存在一个 G 片段或仅有一个 U，C 片段，那么这个链是确定的. 因此，我们将假设至少有两个 G 片段和至少两个 U，C 片段.

我们将使用下面的 G，U，C 片段说明这一过程：

G 片段;CCG,G,UCACG,AAAG,AA

U,C 片段 C,C,GGU,C,AC,GAAAGAA.

我们首先在每一个 G，U，C 之后拆开每一个片段：例如，把片段 GAAAGAA 拆成 G • AAAG • AA，GGU 变成 G • G • U，而 UCACG 变成 U • C • AC • G. 每一部分被称为*尖基部*(extended base)，一个片段中除了第一个和最后一个尖基部外，所有尖基部被称为*内尖基部*(interior extended bases). 使用这个片段的深一层的分解，我们首先观察如何确定所求 RNA 链的开始和结束. 我们做两个列表，一个列表给出由两个分解而形成的所有片段的所有内尖基部，而另外一个列表给出由一个尖基部组成的所有片段. 在我们的例子中，可得到下面的两个列表：

内尖基部,C,C,AC,G,AAAG

由一个尖基部组成的片段:G,AAAG,AA,C,C,C,AC.

不难证明第一个列表中的每一项都在第二个列表中(练习 17). 另外，整个链中的第一个和最后一个尖基部，都在第二列表中而不在第一个列表中(练习 17). 因为我们假设至少有两个 G 片段和两个 U，C 片段，所以不难证明在第二个列表中总是正好存在两项不在第一列表中(练习 17). 其中的一项是这个 RNA 链的第一个尖基部，而另外一项是这个 RNA 链的最后一项. 在我们的例子中，它们是 AA 和 C. 我们如何知道哪个是最后一项? 我们通过观察其中的一个项来自于一个*异常片段*(abnormal fragment)就可做到这一点，即它是一个不结束于 G 的 G 片段的最后一个尖基部，或不结束于 U 或 C 的 U，C 片段的最后一个尖基部. 在我们的例子中，AA 是两个异常片段 AA 和 GAAAGAA 的最后的尖基部. 这表明我们寻找的链开始于 C 并结束于 AA.

645

为了寻找所有可能的链，我们构建一个多重有向图如下：首先确定有多个尖基部的所有正

常片段. 根据每一个这样的片段，使用第一个和最后一个尖基部作为顶点，并画出从第一个尖基部到最后一个尖基部的一条弧，用所对应的片段标记这条弧(我们将增加一条弧). 图 11.31 说明记这一构造. 例如，我们已包括了从 U 到 G 的一条弧，使用所对应的片段 UCACG 的名字标记这条弧. 类似地，我们加入从 C 到 G 和从 G 到 U 的弧. 如果存在多条带有相同的第一个和最后一个尖基部的正常片段，那么也许存在多条从给定的尖基部到另外的尖基部的弧. 最后. 我们把另一条弧加入到我们的多重有向图中. 通过区分出最长的异常片段，并画出从这一异常片段的第一个尖基部到这条链的第一个尖基部的弧，就可以得到这条弧. 这里，这条弧即是 GAAAGAA. 这里我们加入一条从 G 到 C 的弧. 我们用不同的标签标记这条弧，把它记为 $X*Y$，其中 X 是最长的异常片段，$*$ 是标明特殊弧的一个符号，而 Y 是这个链中的第一个尖基部. 因此，在我们的例子中，我们用 GAAAGAA $*$ C 标记从 G 到 C 的弧. 带有给定的 G，U 和 C 片段的每一个可能的 RNA 链都可以从我们构建的多重有向图中区分出来. 可以证明每一个这样的链对应于一条结束于特殊弧 $X*Y$ (练习 20)的一条欧拉封闭路径. 在我们的例子中，唯一的这样的欧拉封闭路径是从 C 到 G 到 U 到 G 到 C. 通过使用对应的弧标签，我们得到下面的链：

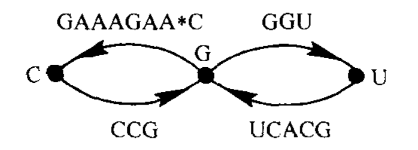


图 11.31　从完全酶分类而得到的 RNA 链的重新构建的多重有向图

CCGGUCACGAAAGAA.

很容易验证这条链有所给的 G，U 和 C 片段. 有关这一图理论方法的更详细的讨论请参见 Hutchinson[1969]. ■

【例 11.2 杂交顺序】⊖ DNA 排序中出现的一个问题是，根据出现在 DNA 串 S 中的所有长度为 k 的子串的列表，确定这个 DNA 串 S 的问题. 称为杂交顺序(sequencing by hybridization)或 SBH 问题，其解决方法再一次使用了欧拉路径的概念(参见 Pevzner[1989]).

考虑 DNA 链中确定的 4 个基 A，C，G，T. 设 L 是串 S 的所有长度为 k 的子串的列表. [这个数据来自于对应于所有 4^k 个长度为 k 的 DNA 序列的合成片段("试样")的 DNA 数组(DNA 碎片). 如果包含一个未知序列的荧光标签 DNA 片段 F 的解被运用于这一数组，那么它与对应于 F 中长度为 k 的子串的试样杂交，其含义是交换 A 和 T 及 G 和 C.] 例如，如果 $S=$ ACACGCAACTTAAA，且 $k=3$，那么

646

$$L=\{\mathrm{ACA},\mathrm{CAC},\mathrm{ACG},\mathrm{CGC},\mathrm{GCA},\mathrm{CAA},\mathrm{AAC},\ \mathrm{ACT},\mathrm{CTT},\mathrm{TTA},\mathrm{TAA},\mathrm{AAA}\}. \tag{11.2}$$

(我们将假设 k 长度子串在 S 中最多出现一次，正如本例子中那样. 实际上，约等于 10 的 k 值足以满足这一目的.)我们如下构建有向图 $D(L)$. 设每一个可能的$(k-1)$长度 DNA 串是 $D(L)$的顶点. 因此 $|V(d(k))|=4^{k-1}$. 对于 L 中的每一个串 $l=b_1b_2\cdots b_k$，在 $D(L)$中存在一条相关的弧 a_l. 弧 a_l 从代表 l 的前 $k-1$ 个基的顶点出发到代表 l 的后 $k-1$ 个基的顶点. 所以，串 l 对应于下面的弧：

$$a_l=(b_1\,b_2\cdots b_{k-1},b_2b_3\cdots b_k).$$

我们还使用 l 的最右边的基 b_k 标记弧 a_l. 最后，我们消除所有 $D(L)$的孤立顶点(参见图 11.32).

如果 P 是 $D(L)$中的一条路径，那么存在按下面方法由 P 构建的相关串 S_P：串 S_P 开始于 P 中的第一个顶点的标签，跟着是 P 的弧上的标签. 例如，在图 11.32 中，路径 $P=$ AC，CG，GC，

⊖ 我们的讨论来自于 Gusfield[1997]和 Pevzner[2000].

CA, AC, CA, AA, AA, AC, CT, TT, TA, AA 产生串 S_P=ACGCACAAACTTAA. 我们称路径 P 指定串 S_P. 有向图 $D(L)$事实上是 11.4.6 节所给出的德布鲁因图表的子图.

给定 k 长度子串的集合 L, 串 S 称为与 L 相容(compatible), 如果 S 包含 L 中的每一个子串且 S 不包含其他长度为 k 的子串. 于是, 目标是寻找与 k 长度 DNA 子串的数据集相容的那些串. 下面定理的证明留作练习 16.

图 11.32 有向图 $D(L)$, 其中 L 是由(11.2)给出的

定理 11.6 *串 S 与 L 相容, 当且仅当 S 由 $D(L)$ 中的一条欧拉路径指定.*

647

根据上面的构造法, 显然与 L 相容的串与 $D(L)$ 中的欧拉路径之间存在一一对应. 所以, 假设 L 中每一个子串在 DNA 串 S 中正好出现一次, 那么 L 唯一确定 S 当且仅当$D(L)$有唯一的欧拉路径. (Van Aardenne Ehrenfest 和 de Bruijn[1951]给出了有向图 D 的欧拉路径的数量公式.)

完成我们的例子, 注意, 图 11.32 的有向图 $D(L)$中的每一个顶点的入度数等于出度数, 只有两个顶点 AC, AA 除外. 对于这两个顶点, 出度数(AC)=1+入度数(AC), 而入度数(AA)=1+出度数(AA). 因此, $D(L)$有一条欧拉路径.

尽管根据 k 长度子串重新构建 DNA 序列的目标似乎在理论上是可达到的, 然而杂交顺序并不像我们所描述的那样简单. 根据一个未知的子串 S 生成 L 中的子串的过程中通常会出现错误. 同时, 我们对 L 中的子串不重复的假设不是非常合理, 这与 DNA 串的性质有关. 对于更多的杂交顺序, 请参见 Drmanac 等[1989]、Gusfield[1997]、Pevzner[2000]或 Pevzner 和 Lipshutz [1994]. ■

11.4.5 编码应用

Hutchinson 和 Wilf[1975]在下面的假设下研究了有 n 个字母的字母表上的代码: 代码字中的所有信息都由每一个类型的字母数量以及字母的序对的频率所携带, 即由一个字母跟在另一个字母之后的频率所携带. 例如, Hutchinson 和 Wilf 把 DNA 或 RNA 分子当作一个字来处理, 且以基(不是尖基部)作为字母, 并对基因代码作上面的假设. 特别地, Hutchinson 和 Wilf 提出下面的问题: 给定的非负整数 v_i, v_{ij} $(i, j=1, 2, \cdots, n)$是否存在有 n 个字母的字母表中的字, 使得第 i 个字母正好出现 v_i 次, 并使得第 j 个字母跟随第 i 个字母之后的次数正好是 v_{ij} 呢? 如果存在这样的字, 那么所有这样的字是什么呢? 我们将给出 Hutchinson 和 Wilf 的解, 这个解使用了欧拉路径的概念.

作为例子, 假设 $v_1=2$, $v_2=v_3=1$, 且 v_{ij} 是由下面的矩阵给出的:

$$(v_{ij})=\begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix}. \tag{11.3}$$

这时, 如果 A 对应于第一字母, B 对应于第二个字母, C 对应于第三个字母, 那么有指定模式的一个字是 $ABCA$. 再给一个例子, 假设 $v_1=2$, $v_2=4$, $v_3=3$ 且有下面的矩阵:

$$(v_{ij})=\begin{bmatrix}0&0&2\\2&1&1\\0&2&0\end{bmatrix}. \tag{11.4}$$

648

那么有指定模式的一个字是 BBCBACBAC.

为了分析我们的问题，让我们画出以 n 个字母 A_1，A_2，…，A_n 为顶点且从 A_i 到 A_j 有 v_{ij} 个弧的多重有向图. 允许有环. 对应于矩阵(11.3)和矩阵(11.4)的多重有向图分别如图 11.33 和图 11.34 所示. 假设 $w=A_{i_1}$，A_{i_2}，…，A_{i_q} 是一个构成解的字. 那么，显然 w 对应于多重有向图中开始于 A_{i_1} 并结束于 A_{i_q} 的一条欧拉路径. 对于构成我们给出的两个例子的两个解的字很容易看到这一点. 在下文中，我们使用构成解的字对应于一条欧拉路径的观察结果，了解更多关于构成解的字. 读者可以跳过本小节余下的内容.

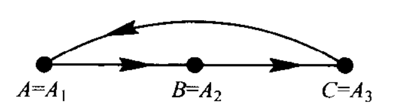


图 11.33　对应于(11.3)式的多重有向图

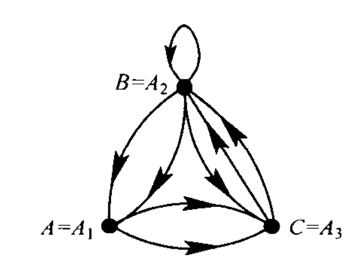


图 11.34　对应于(11.4)式的多重有向图

因为一个构成解的字对应于一条欧拉路径，所以如果存在一个构成解的字，那么 D 必定是相对于孤立顶点弱连通的. 我们首先考虑 $i_1 \neq i_q$ 的情况. 对于每一个 $i \neq i_1$，i_q，我们在 A_i 处的入度数等于出度数. 对于 $i=i_1$，A_i 的出度数比入度数大 1，且对于 $i=i_q$，A_i 的入度数比出度数大 1. 因此，我们有

$$\sum_{k=1}^{n} v_{ik} = \begin{cases} \sum_{k=1}^{n} v_{ki}, & i \neq i_1, i_q \\ \sum_{k=1}^{n} v_{ki} + 1, & i = i_1 \\ \sum_{k=1}^{n} v_{ki} - 1, & i = i_q. \end{cases} \tag{11.5}$$

这个条件说明在矩阵(v_{ij})中，行的和等于所对应的列的和，除了在两个位置上它们以标明的方式相差 1 之外. 我们还有一个一致性条件，这个条件给出了 v_i 和 v_{ij} 的关系：

$$v_i = \begin{cases} \sum_{k=1}^{n} v_{ki}, & i \neq i_1 \\ \sum_{k=1}^{n} v_{ki} + 1, & i = i_1. \end{cases} \tag{11.6}$$

使用定理 11.5，容易看到，如果条件(11.5)和条件(11.6)对于某两个 i_1 和 i_q($i_1 \neq i_q$)成立，且如果 D 是相对于孤立顶点弱连通的，那么存在一个构成解的字，而且每一个构成解的字对应于一条开始于 A_{i_1} 并结束于 A_{i_q} 的欧拉路径. 在我们的第二个例子中，条件(11.5)和条件(11.6)对于 $i_1=2$，$i_q=3$成立. 存在若干条从 $B=A_{i_1}$ 到 $C=A_{i_q}$ 的欧拉路径，而且每一条欧拉路径给出一个构成解的字. 第二个例子是 BACBBCBAC. [649]

如果这个构成解的字开始且结束于相同的字母，即如果 $i_1=i_q$，那么情况又如何呢? 这时，存在一条欧拉封闭路径，且我们有

$$\sum_{k=1}^{n} v_{ik} = \sum_{k=1}^{n} v_{ki}, \quad i = 1,2,\cdots,n. \tag{11.7}$$

同样，给定 i_1，(11.6)式(对于所有 i)成立. 条件(11.7)表明在(v_{ij})中，每一个行的和等于其对应列的和. 相反，如果(11.7)式成立，那么对于 i_1，(对于所有的 i)条件(11.6)成立，且如果 D 是相对于孤立顶点弱连通的，那么存在一个解，且每一个构成解的字对应于多重有向图 D 中的

一条欧拉封闭路径，这条欧拉封闭路径开始并结束于 A_{i_1}. 这就是在我们的第一个例子中 $i_1=1$ 时的解.

总之，如果存在一个构成解的字，那么 D 相对于孤立顶点弱连通，且对于某个 i_1(11.6)式成立. 另外，或者(11.7)式成立，或者对于 i_1 和某个 i_q，$i_1\neq i_q$，(11.5)式成立. 相反，假设 D 相对于孤立顶点弱连通. 如果对于某个 i_1，(11.6)式成立，且对于 i_1 和某个 i_q，$i_1\neq i_q$，(11.5)式成立，那么存在一个解，且所有构成解的字对应于开始于 A_{i_1} 且结束于 A_{i_q} 的一条欧拉路径. 如果对于某个 i_1，(11.6)式成立，且(11.7)式成立，那么存在一个解，而且所有解的字对应于开始且结束于 A_{i_1} 的一条欧拉封闭路径.

11.4.6　德布鲁因序列和电信

本小节考虑另外一个编码问题及其在电信方面的应用. 设

$$\Sigma=\{0,\ 1,\ \cdots,\ p-1\}$$

是有 p 个字母的字母表，考虑 Σ 上所有长度为 n 的字. $(p,\ n)$德布鲁因序列(是以 N. G. de Bruijn 的名字命名的)是序列

$$a_0a_1\cdots a_{L-1}. \tag{11.8}$$

其中，每一个 a_i 是 Σ 中的字母，且对于 Σ 上长度为 n 的每一个字 w，正好存在一个 i，使得 w 有如下形式

$$a_ia_{i+1}\cdots a_{i+n-1}, \tag{11.9}$$

其中，(11.9)式中的下标的加法是模 L 的[⊖]. 例如，

$$01110100$$

是字符表{0, 1}上的一个(2, 3)德布鲁因序列，其中 $n=3$. 因为从序列的起点开始，可以分别得到如下 3 字母字 $a_ia_{i+1}a_{i+2}$：

$$011,111,110,101,010,100,000,001.$$

最后的两个 3 字母字是 $a_6a_7a_0$ 和 $a_7a_0a_1$. 德布鲁因序列在编码理论中意义很大. 可以通过移位寄存器，有时候又称为*移位寄存器序列*(shift register sequence)来实现它们. 有关这一课题的更详细论述请参见 Golomb[1982].

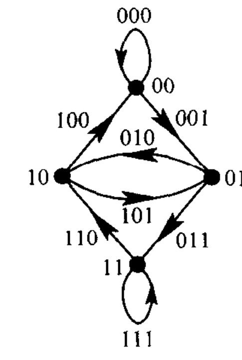


图 11.35　有向图 $D_{2,3}$

显然，如果存在$(p,\ n)$德布鲁因序列(11.8)，L 必定是 p^n，其中 $p=|\Sigma|$. 我们将证明对于每一对正整数 p 和 n，存在$(p,\ n)$德布鲁因序列(11.8). 给定 p 和 n，如下构建称为*德布鲁因图表*(de Bruijn diagram)的有向图 $D_{p,n}$. 设 $V(D_{p,n})$由字母表 Σ 上所有长度为 $n-1$ 的字组成，且包含从字 $b_1b_2\cdots b_{n-1}$ 到每一个形如 b_2，$b_3\cdots b_n$ 的字的弧. 以字 $b_1b_2\cdots b_n$ 标记这样的弧. 图 11.35 和图 11.36 给出 $p=2$，$n=3$ 和 $p=3$，$n=2$ 时的有向图 $D_{p,n}$. 假设德布鲁因图表有一条欧拉封闭路径. 这时，我们连续使用这条路径中的每一条弧的第一个字母，得到一个德布鲁因序列(11.8)，其中 $L=p^n$. 为了清楚看到这是德布鲁因序列，注意我们可以识别 Σ 上长度为 n 的每一个字 w. 因为如果 $w=b_1b_2\cdots b_n$，我们知道这条欧拉路径覆盖从 $b_1b_2\cdots b_{n-1}$ 到 $b_2b_3\cdots b_n$ 的弧. 因此，这条欧拉路径必定从 $b_1b_2\cdots b_{n-1}$ 到 $b_2b_3\cdots b_n$. 从这里开始，它必定是从 $b_3b_4\cdots$ 到 $b_4b_5\cdots$ 到 $b_n\cdots$. 因此，这条路径中的弧的前几个字母是

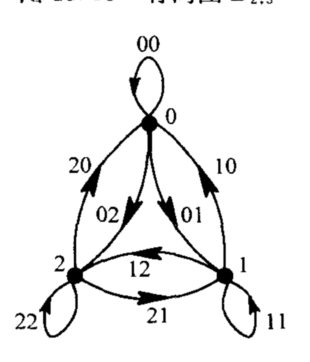


图 11.36　有向图 $D_{3,2}$

⊖ 关于模算术的讨论，参见 9.3.1 节.

$b_1b_2b_3\cdots b_n$. 利用定理 11.4，容易看出，图 11.35 的德布鲁因图表有一条欧拉封闭路径，因为这个有向图是弱连通的，且每一个顶点的入度数等于出度数. 德布鲁因序列 01110100 对应于从 01 到 11 到 11 到 10 到 01 到 10 到 00 到 00 到 01 的欧拉封闭路径.

我们将证明下面的定理，这一定理表明对于每一对正整数 p 和 n，存在(p, n)德布鲁因序列. 对于 $p=2$ 的情况，这一定理是由德布鲁因发现的，对于任意的 p 的情况，这一定理是由 Good[1946]发现的.

651

定理 11.7　*对于所有正整数 p 和 n，德布鲁因图表 $D_{p,n}$有欧拉封闭路径.*

证明　首先，我们要证明 $D_{p,n}$是弱连通的，事实上它是强连通的. 设 $b_1b_2\cdots b_{n-1}$ 和 $c_1c_2\cdots c_{n-1}$是 $D_{p,n}$的任意两个顶点. 因为我们有如下路径：

$$b_1b_2\cdots b_{n-1},\quad b_2b_3\cdots b_{n-1}c_1,\quad b_3b_4\cdots b_{n-1}c_1c_2,\cdots,\quad c_1c_2\cdots c_{n-1},$$

所以 $D_{p,n}$是强连通的. 接下来，注意每一个顶点的入度数和出度数都等于 p. 所以根据定理 11.4，此定理成立. ■

通过把我们的结果运用于电信中的一个问题来结束本小节. 我们按照 Liu[1968]. 一个滚筒有 8 个不同的扇片. 问题是：我们能否不观察这个滚筒就能知道它的位置？一个方法是在某些扇片上放置导电材料，而在另外一些扇片上放置非导电材料. 把三个电极相邻地放置到滚筒旁，使得在这个滚筒的任意位置中，电极都邻接三个连续的扇片，如图 11.37 所示. 如果电极邻接有导电材料的扇片，它将会被触发. 如果我们更聪明点，那么我们将选择这样的导电和非导电材料模式，使得触发的和未触发的电极模式能告知我们这个滚筒的位置.

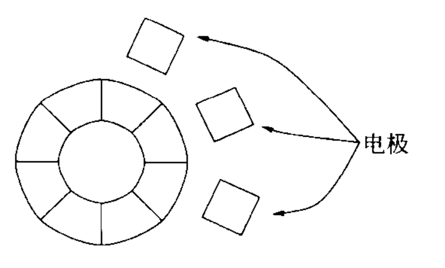


图 11.37　带有 8 个扇片和 3 个邻接电极的滚筒

我们可以重新阐述这一事实如下. 设每一个扇片得到一个 1 或一个 0. 我们希望把 8 个 0 和 1 绕圆分布，使得三个连续数字的每一个序列不同. 通过寻找一个(2, 3)德布鲁因序列，特别是序列 01110100，就可以做到这一点. 如果我们如图 11.38 那样在一个圆的周围布局这些数据，那么下面的连续数字的序列将从箭头开始，以逆时针顺序出现：011，111，110，101，010，100，000，001. 这些三数字项，如我们希望的那样都是不同的. 因此，可以唯一编码滚筒的每一个位置.

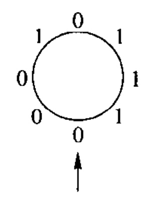


图 11.38　求解滚筒问题的 0，1 分布

652

与电传打印和密码学相关的问题将在练习 22 和练习 23 中给以描述.

本节练习

1. 在图 11.28 的图 G 中，寻找另一种方法来加入 G 的边的 4 个拷贝，以得到有欧拉闭链的多重图.
2. 枚举出图 11.32 中的有向图 $D(L)$中的所有欧拉路径.
3. 在图 11.39 的每一个图中，通过寻找能够产生有欧拉闭链的多重图的 G 的边的最小拷贝数量，寻找从 x 到 x 的邮递员最短路径.
4. 给定图 11.40 的清扫子图和马路多重有向图，确定加入到清扫子图的空载返回弧的集合，以得到有欧拉封闭路径的多重有向图. 确定经过这条路径，包括清扫和空载返回时间在内的所需时间总量.
5. 对于下面各组数字，画出德布鲁因图表 $D_{p,n}$并寻找(p, n)德布鲁因序列：
   (a)$p=2$，$n=4$　　　　(b)$p=3$，$n=3$
6. 对于下列 2.12 节各练习中所给定的 G，U，C 片段，使用 11.4.4 节的方法确定 RNA 链.

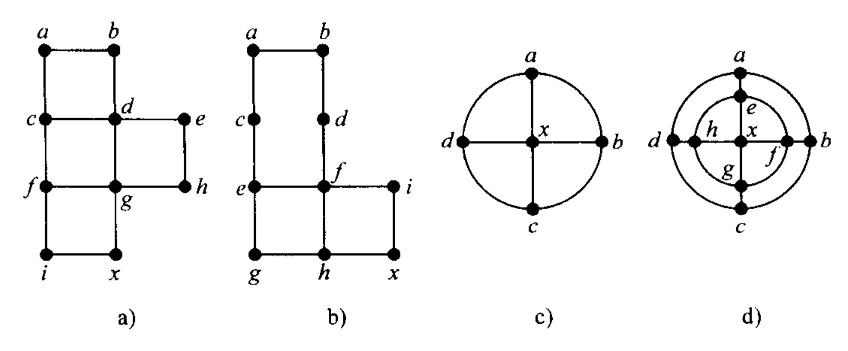


图 11.39　11.4 节练习 3 的图

(a)练习 1　　　　(b)练习 3

(c)练习 5

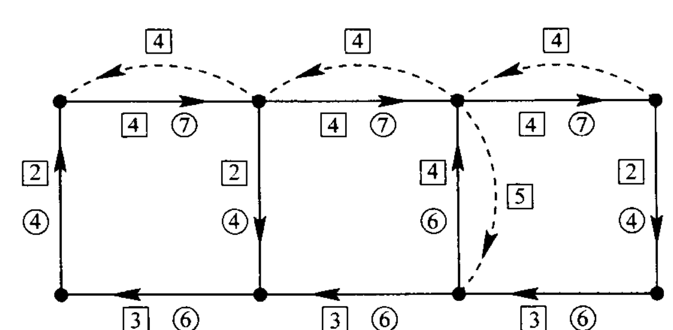


图 11.40　带有清扫子图的马路多重有向图以及如图 11.29所示的清扫和空载返回时间

7. (a)对于列表 $L=\{$CCGA，CGCG，CGTG，CGAA，GCCG，GCGT，GTGC，GAAC，TGCC，AACC，ACCC$\}$，画出例 11.2 中的有向图 $D(L)$.

(b)确定 $D(L)$中的欧拉路径的数量.

(c)确定 L 的相容串(s).

8. (a)对于列表 $L=\{$AA，AC，AG，AT，CA，CT，GA，GT，TA，TC，TG，TT$\}$，画出例 11.2 中的有向图 $D(L)$.

(b)确定 $D(L)$的欧拉路径的数量.

(c)$S=$ACTTCATGAGTAA 与 L 相容吗?

(d)$S=$AATGAGTACTTCA 与 L 相容吗?

(e)$S=$CTAGTACATGATA 与 L 相容吗?

9. 对于下列每一种情况，确定是否存在这样的码字：它的第 i 个字母正好出现 v_i 次，第 j 个字母跟随第 i 个字母后面的次数正好是 v_{ij}，如果存在这样的码字寻找一个这样的字.

(a)$v_1=3$，$v_2=4$，$v_3=5$，

$$(v_{ij})=\begin{bmatrix}1&1&0\\1&2&1\\0&1&4\end{bmatrix}.$$

(b)$v_1=3$，$v_2=4$，$v_3=5$，

$$(v_{ij})=\begin{bmatrix}1&1&1\\1&2&1\\0&1&3\end{bmatrix}.$$

(c)$v_1=3$，$v_2=4$，$v_3=5$，

$$(v_{ij})=\begin{bmatrix}1&1&1\\1&2&1\\1&1&3\end{bmatrix}.$$

(d)$v_1=3$，$v_2=3$，$v_3=3$，$v_4=3$，

$$(v_{ij})=\begin{bmatrix}1&1&0&1\\1&1&1&1\\1&0&1&1\\0&1&1&1\end{bmatrix}.$$

(e)$v_1=3$，$v_2=3$，$v_3=2$，$v_4=3$ 且(v_{ij})如部分(d).

10. 对于"中国邮差"问题，证明如果 x 和 y 是 G 的任意两个顶点，从 x 到 x 的邮递员的最短路径与从 y 到 y 的邮递员的最短路径的长度相同.

653～654

11. 注意，在图 11.28 的第二个多重图中，虚线的边可以分成两条链，每一条链连接 G 中的两个奇度顶点. 这些链是 d，e，b 和 h，x，f.

    (a)在图 11.28 的第三个多重图中能够寻找到两条这样的链吗?

    (b)一般地，如果 H 是从 G 通过使用邮递员最短路径中的边而得到的多重图，证明 H 中的虚线边可以分类成若干条链(不一定是两条)，每一条链连接 G 中的奇度数的两个顶点. 而且，G 中的每一个奇度数的顶点正好是其中一条链的端顶点(这一结果在 12.7.1 节中讨论的"中国邮差"问题的最后部分的解很重要).

12. 在一条邮递员最短路径中，存在可以使用两次以上的任意边吗? 为什么?

13. 假设对于街道清扫问题我们有最优解，即在马路多重有向图中有一条封闭路径，这条路径包括清扫子图的所有弧，而且使用尽可能少的时间总量. 是否有可能某条弧被经过的次数超过两次?

14. 给定弱连通有向图 D，是否总存在至少使用 D 的每条弧一次的路径呢?

15. 证明有向图 D 是强连通的，当且仅当 D 有一个至少使用 D 的每一条弧一次的回路.

16. 证明定理 11.6.

17. 证明下面关于 G，U，C 片段的内部尖基部列表，及由一个尖基部组成的片段列表的事实.

    (a)第一个列表上的每一项都在第二个列表中.

    (b)完整的 RNA 链中的第一个和最后一个尖基部在第二个列表中，但不在第一个列表中.

    (c)如果至少存在两个 G 片段和至少两个 U，C 片段，那么在第二个列表中正好总存在两个不在第一个列表中的项.

18. 在什么样的环境下，一个 RNA 链有两个异常片段?

19. 确定最小可能的歧义 RNA 链的长度，即这个 RNA 链的 G，U，C 片段与另一个 RNA 链的 G，U，C 片段相同.

20. 本练习概括给出断言 S 的证明[㊀]：假设给定 RNA 链的 G 和 U，C 片段，该链至少有两个 G 片段和至少有两个 U，C 片段. 那么有给定的 G 和 U，C 片段的每一个 RNA 链，对应于一条结束于 11.4.4 节构造的多重有向图 D 中的弧 $X*Y$ 的一条欧拉封闭路径. 我们说一个尖基部有*类型*(type)G，如果它结束于 G，有*类型*(type)U，C，如果结束于 U 或 C. 注意，任意的 RNA 链都可以写作下面的形式：

$$A_0A_1\cdots A_kA_{k+1}B,$$

其中每一个 A_i 都是由一种类型的尖基部组成的，A_i 的类型交替改变，而且如果这个链结束于 A，则 B 是最后一个尖基部，否则 B 是空链. 对于 $i=0, 1, \cdots, k$，设 $\overline{A}_i$ 是 A_ia_{i+1}，其中 a_j 是 A_j 中的第一个尖基部. 设 $\overline{A}_{k+1}$ 是 $A_{k+1}B$. 最后，称一个尖基部的一个片段是*平凡的*(trivial). 证明下面的事实.

655

    (a)$\overline{A}_0$，$\overline{A}_1$，$\cdots$，$\overline{A}_k$ 都是非平凡正常片段，且不存在其他非平凡正常片段.

    (b)$\overline{A}_{k+1}$ 是最长的异常片段.

    (c)a_0，a_1，$\cdots$，a_{k+1} 是多重有向图 D 的顶点，$\overline{A}_i$ 是从 a_i 到 a_{i+1} 的弧上的标签，且 $\overline{A}_{k+1}*a_0$ 是从 a_{k+1} 到 a_0 的弧上的标签.

    (d)标签为 $\overline{A}_0$，$\overline{A}_1$，$\cdots$，$\overline{A}_{k+1}$，$\overline{A}_{k+1}*a_0$ 的弧按此顺序定义一条欧拉封闭路径.

    (e)产生相同非平凡正常片段集合和相同的最长异常片段的任何其他 RNA 链，产生相同的多重有向图 D.

    (f)由 D 构建的任意 RNA 链产生相同的非平凡正常片段集合和相同的最长异常片段.

    (g)非平凡正常片段和最长异常片段唯一确定所有片段.

㊀ 对于支持这一证明的思想，作者要感谢 Michael Vivino.

(h)陈述 S 成立.

21. 在 11.4.5 节所讨论的问题中，如果存在构成解的字，那么每一条欧拉路径正好确定一个构成解的字. 然而，每一个构成解的字可以确定多少条不同的欧拉路径呢？
22. 下面的问题出现在密码学中：从给定的有 m 个字母的字母表确定一个字，在这个字中 r 个字母的每一个排列正好出现一次. 如果 $r=3$，$m=4$，确定这一问题的解.
23. 通信中一个被称为电传打印的重要问题如下：由 0 和 1 构成的使得所有的 r 个连续位序列的出现次数不超过一次的最长环形序列是多长？(当 r 位序列在末端附近开始并结束于开始端时，我们(也)认为它出现.)求解这个电传打印问题(这个问题是由 Good[1946]最先解决的).

11.5 哈密顿链和哈密顿路径

11.5.1 定义

在图或有向图中，类似于欧拉链或路径的是*哈密顿链或路径*(hamiltonian chain or path)，这样的链或路径使用每一个顶点一次且仅一次. 哈密顿链的概念要追溯到 William Rowan Hamilton，他于 1857 年描述了导致这一概念的一个谜题(参见下面). 本节讨论哈密顿链和路径的存在问题. 这里我们讨论其中的一些应用并在 11.6 节中讨论另外一些应用.

注意，哈密顿链或路径本身就是简单链或路径，因此根据我们的习惯它也不能是封闭的. 然而，我们称一个*回路或循环* $u_1, u_2, \cdots, u_t, u_1$ 是*哈密顿的*(hamiltonian)，如果相应的 $u_1, u_2, \cdots, u_t$ 是哈密顿链或路径.

尽管哈密顿链、路径等概念与相应的欧拉链和路径的概念类似，但是我们很难知道一个图或有向图是否存在这样的对象. 实际上，确定一个图或有向图是否有哈密顿链(路径、回路、循环)是 NP 完全问题. 关于这些对象的存在问题已知一些结果，我们在这里给出其中的若干结果. 首先我们给出哈密顿链、路径、回路或循环的一些应用.

656

【例 11.3 跟踪十二面体的边】 我们从一个例子开始. 哈密顿谜题是确定跟踪十二面体的边的方法，该方法访问每一个角正好一次并返回到出发点. 把十二面体的顶点和边画成如图 11.41 所示的图. 这个问题就变成在这个图中确定一条哈密顿回路. 存在一条这样的回路：$a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, a$. ■

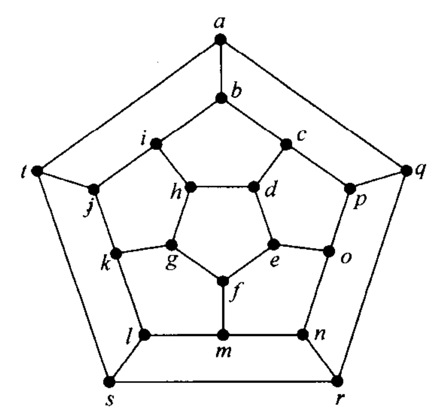


图 11.41 表示正十二面体的顶点和边的图

【例 11.4 卖货郎问题(再探例 2.10)】 哈密顿循环的概念也出现在卖货郎问题中，这一问题我们已在 2.4 节中讨论过. 一个卖货郎想要访问 n 个不同城市，且每一个城市正好访问一次，然后返回到他的出发地点，其目的是最小化成本(在 2.4 节中，我们讨论了卖货郎问题的一些其他应用，例如，寻找银行信使的最优路径，或寻找自动仓库中机器人的最优路径). 假设我们设各城市是一个完全对称有向图的顶点，有向图的所有顶点由两条弧连接，并假设如果 c_{ij} 是从城市 i 到城市 j 旅途所花费的成本，则我们在弧(i, j)上放置权 c_{ij}. 因为这个完全对称有向图有哈密顿循环，所以存在问题没有意义. 相反我们要在这个有向图中寻找一条权的和为最小的哈密顿循环. 正如我们前面所指出的那样，卖货郎问题很难，它是一个 NP 完全问题. ■

657

【例 11.5 调度生产过程】 一家制造工厂仅有单一的加工设备. 很多产品要在这里加工. 如果产品 j 在产品 i 之后立即加工，那么从加工产品 i 的配置到加工产品 j 的配置，重新设置加工

设备的成本是 c_{ij}. 如果没有必要重新设置，那么，成本当然是 0. 假设加工产品的成本与产品的加工顺序无关，那么这一问题就是选择一种顺序，使得重新设置的成本 c_{ij} 的总和最小. 正如我们在例 2.17 中所观察的那样，类似的问题也出现在调度计算机运行，以及化工和制药工业中，在那里，加工设备可能是反应器且重新设置意味着需要清扫. 当然，如果我们可以寻找到一种生产调度，在这一调度中没有必要重新设置设备，那么成本就可以达到最小. 为了弄明白这样的生产调度是否存在，我们构建一个有向图 D，这个图的顶点是待加工的产品，如果产品 j 在产品 i 之后加工且不必重新设置设备，那么这个有向图有从 i 到 j 的一条弧. 如果在 D 中存在哈密顿路径，那么存在这样的生产调度. 关于这一问题的更详细内容以及对不存在哈密顿路径情况的论述等相关内容请参见 Christofiede[1975]. 还可以参考 11.6.3 节的相关讨论. ■

11.5.2 图存在哈密顿回路的充分条件

并不是每个图都有哈密顿回路. 考虑图 11.42 中的图. 我们注意到，这个图中的每一条边把 $A=\{a, b, c\}$中的一个顶点与 $B=\{x, y\}$中的一个顶点连接起来. 因此，哈密顿回路就必须从 A 中的顶点到 B 中的顶点交错通过. 如果 $|A|=|B|$，这种情况可以发生.

图 11.42 没有哈密顿回路的图

本小节给出哈密顿回路存在的充分条件.

定理 11.8(Ore[1960]) *假设 G 是有 n 个顶点，且 $n\geq 3$ 的图，而且只要 $x\neq y$ 的顶点没有被边连接，那么 x 的度数加上 y 的度数总是至少等于 n. 这时，G 有哈密顿回路.*

证明[⊖] 假设 G 没有哈密顿回路. 我们将证明对于 $V(G)$中的某些不邻接的 x，y，有

658

$$\deg_G(x)+\deg_G(y)<n, \tag{11.10}$$

其中 $\deg_G(a)$表示 a 在 G 中的度数. 如果我们增加 G 的边，那么最终我们可以得到一个完全图，这个完全图有哈密顿回路. 因此，在增加边的过程中，我们必定最后会遇到有下面性质的图 H：H 没有哈密顿回路，但是在 H 中增加一条边，可以给出有哈密顿回路的图. 我们将证明在 H 中，不邻接的 x 和 y 使得

$$\deg_H(x)+\deg_H(y)<n. \tag{11.11}$$

但是，对于所有 a，有 $\deg_G(a)<\deg_H(a)$，所以(11.11)式蕴涵(11.10)式.

在 H 中选择任意不邻接的 x 和 y. 然后在 H 中加上边$\{x, y\}$就有哈密顿回路. 因为 H 没有哈密顿回路，所以这条哈密顿回路必须使用边$\{x, y\}$. 因此，这条回路可以写成

$$x,y,a_1,a_2,\cdots,a_{n-2},x.$$

现在，$V(H)=\{x, y, a_1, a_2, \cdots, a_{n-2}\}$. 另外，我们注意到对于 $i>1$，有

$$\{y,a_i\}\in E(H)\Rightarrow\{x,a_{i-1}\}\notin E(H). \tag{11.12}$$

因为如果上式不成立，那么就有

$$y,a_i,a_{i+1},\cdots,a_{n-2},x,a_{i-1},a_{i-2},\cdots,a_1,y$$

是 H 中的哈密顿回路，这是矛盾的. 现在，(11.12)式和边$\{x, y\}\notin E(H)$蕴涵(11.11)式. ■

为了说明 Ore 定理，注意图 11.43 中的图 a 有哈密顿回路，因为任意两个顶点的度数和至少等于 5. 注意 Ore 定理没有给出必要条件. 因为考虑长度为 5 的回路 Z_5. 它存在哈密顿回路，但是任意两个顶点的度数和等于 4，这小于 n.

⊖ 此证明可以跳过.

下面的结果，原本是独立证明的，但是它是定理 11.8 的直接结果.

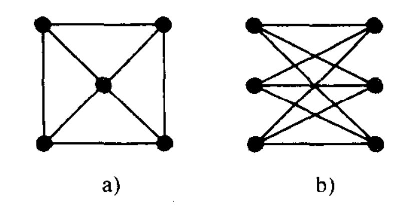


图 11.43 有哈密顿回路的两个图

659

推论 11.8.1(Dirac[1952]) *假设 G 是有 n 个顶点的图且 $n\geqslant 3$，每个边的度数至少等于 $n/2$. 那么 G 有哈密顿回路.*

为了说明推论 11.8.1，注意图 11.43b 有一条哈密顿回路，因为它的每一个顶点的度数为 3，且有 6 个顶点.

推论 11.8.2(Bondy 和 Chvátal[1976]) *假设 G 是有 n 个顶点的图且 $n\geqslant 3$，x 和 y 是 G 中不邻接的顶点，使得*

$$\deg(x)+\deg(y)\geqslant n.$$

那么 G 有哈密顿回路当且仅当 G 加上边 $\{x, y\}$ 有哈密顿回路时.

证明 如果 G 有哈密顿回路，那么 G 加上边 $\{x, y\}$ 肯定有哈密顿回路. 逆叙可以从定理 11.8 的证明得到. ■

我们看一下当我们尝试着重复推论 11.8.2 的构造法时会发生什么. 也就是说，我们从图 $G=G_1$ 开始. 我们在 G_1 中寻找一对不邻接的顶点 x_1 和 y_1，使得在 G_1 中 x_1 和 y_1 的度数的和至少等于 n. 设 G_2 是 G_1 加上边 $\{x_1, y_1\}$. 现在，我们在 G_2 中寻找一对不邻接的顶点 x_2 和 y_2，使得在 G_2 中 x_2 和 y_2 的度数之和至少等于 n. 设 G_3 是 G_2 加上边 $\{x_2, y_2\}$. 我们继续这一过程直到我们得到图 $H=G_i$，且没有在 G_i 中的度数之和至少等于 n 的不邻接的顶点 x_i 和 y_i. 不难证明(练习 17)无论我们执行构造的顺序如何，我们总是可以得到相同的图 H. 我们称这样的图 H 是 G 的闭包(closure)，并把它记为 $c(G)$. 我们用图 11.44 说明这一构造法. 下面的结果是从推论 11.8.2 得到的.

推论 11.8.3(Bondy 和 Chvátal[1976]) *G 有哈密顿回路当且仅当 $c(G)$ 有哈密顿回路.*

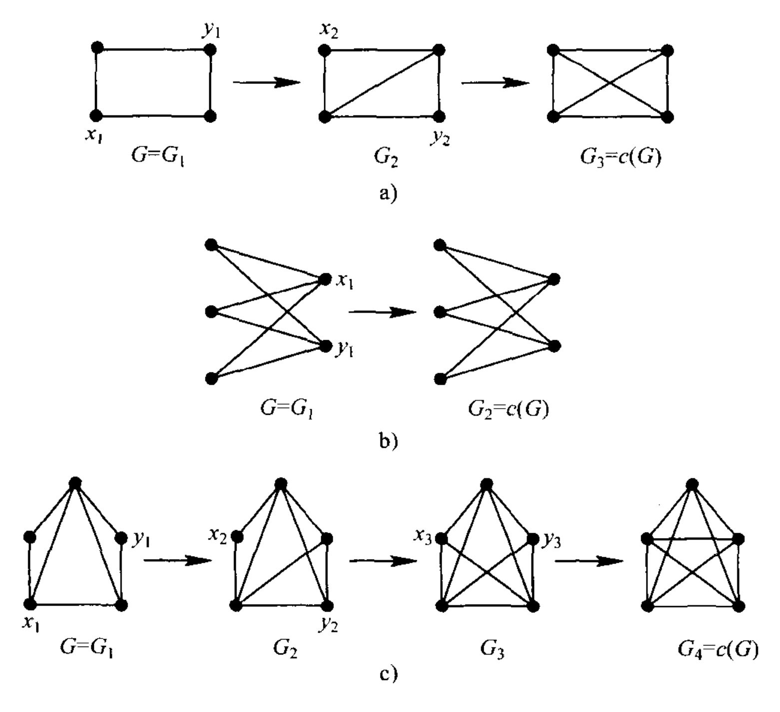


图 11.44 三个闭包的构造

注意，在图 11.44 中，在图 a 和图 c 中 $c(G)$是完全图，而在图 b 中不是. 图 a 和图 c 说明下面的定理. [注意图 c 不能使用 Ore 定理来处理.]

定理 11.9(Bondy 和 Chvátal[1976])　假设 G 是有至少三个顶点的图. 如果 $c(G)$是完全的，那么 G 有哈密顿回路.

证明　至少有三个顶点的完全图有哈密顿回路. ■

注意，Ore 定理是定理 11.9 的推论.

我们给出了哈密顿回路存在的充分条件，而没有给出哈密顿链存在的充分条件. 对于后者存在的一些充分条件，可以参考 Chartrand 和 Lesniak[1996]的内容和练习.

11.5.3　有向图存在哈密顿循环的充分条件



Ore 和 Dirac 的结果(定理 11.8 和推论 11.8.1)对于有向图有下面类似的结果. 这里，$\mathrm{od}(u)$是 u 的出度数，而 $\mathrm{id}(u)$是 u 的入度数.

定理 11.10(Woodall[1972])　假设 D 是有 n 个顶点的有向图，$n \geqslant 3$，而且只要 $x \neq y$，且不存在从 x 到 y 的弧，则

$$\mathrm{od}(x) + \mathrm{id}(y) \geqslant n, \tag{11.13}$$

那么 D 有哈密顿循环.

定理 11.11(Ghouila-Houuri[1960])　假设 D 是有 n 个顶点的强连通有向图，且对于每一顶点 x 有

$$\mathrm{od}(x) + \mathrm{id}(x) \geqslant n \tag{11.14}$$

那么 D 有哈密顿循环.

推论 11.11.1　假设 D 是有 n 个顶点的有向图，且对于每一个顶点 x 有

$$\mathrm{od}(x) \geqslant n/2 \quad 和 \quad \mathrm{id}(x) \geqslant n/2, \tag{11.15}$$

那么 D 有哈密顿循环.

证明　我们可以证明(11.15)式蕴涵 D 是强连通的. 证明留给读者(练习 18). ■

我们给出了哈密顿循环存在的充分条件，但没有给出哈密顿路径存在的充分条件. 对于后者存在的充分条件的概括，请参考 Chartrand 和 Lesniak[1996].

本节练习

1. 对于图 11.45 中的每一个图，寻找一条哈密顿回路.

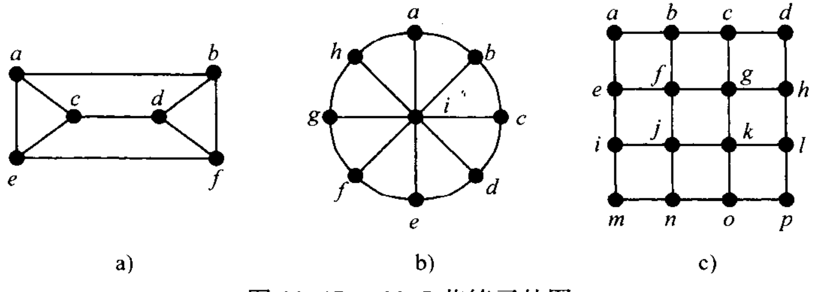


图 11.45　11.5 节练习的图

662

2. 对于图 11.46 中的每一个图，寻找一条哈密顿链.
3. 对于图 11.47 中的每一个图，寻找一条哈密顿循环.
4. 对于图 11.48 中的每一个图，寻找一条哈密顿路径.
5. 证明图 11.49 中的图没有哈密顿回路. [提示：a 和 x 将告诉你一些事情.]
6. 图 11.50 的图有哈密顿回路吗？为什么？
7. 给出满足下面条件的图的例子：

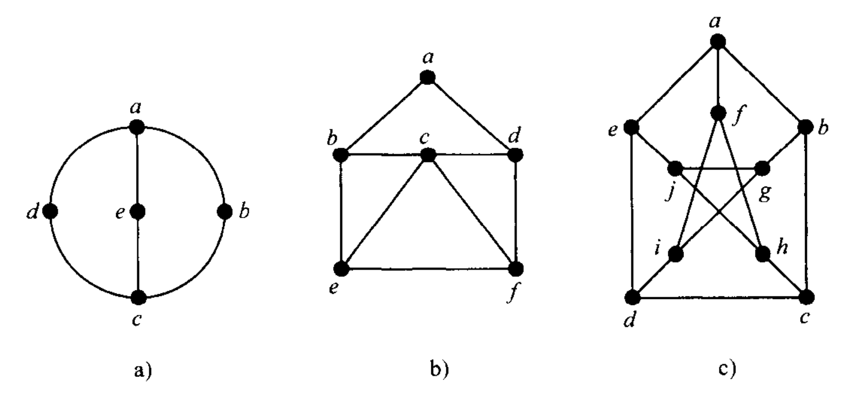


图 11.46　11.5 节练习的图

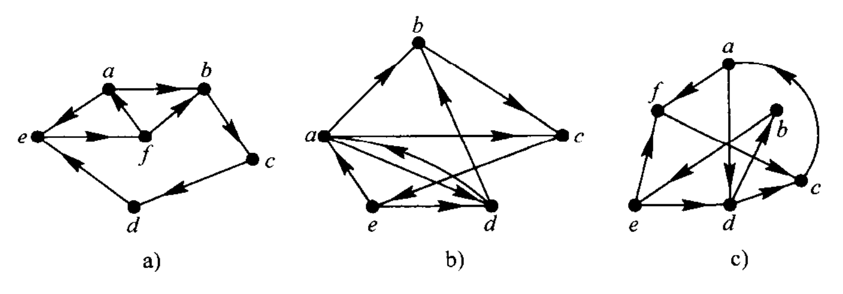


图 11.47　11.5 节练习的有向图

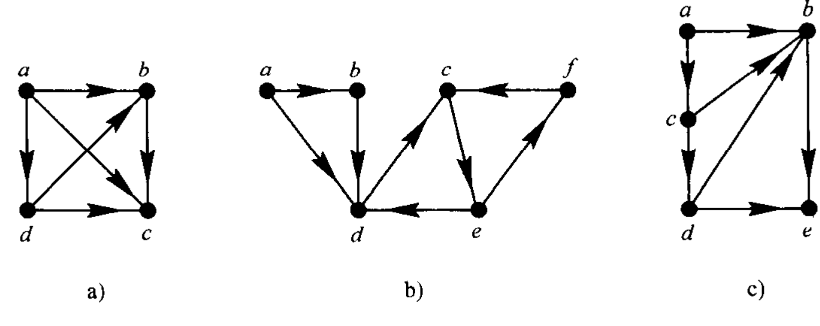


图 11.48　11.5 节练习的有向图

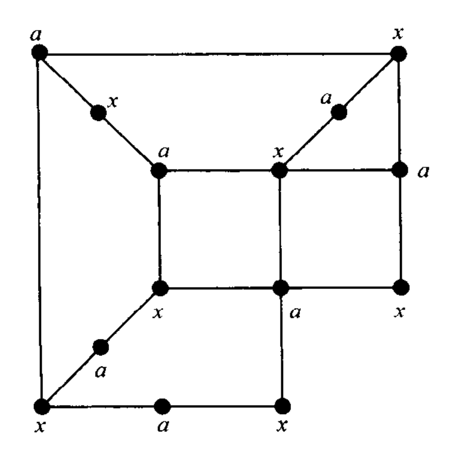


图 11.49　没有哈密顿回路的图

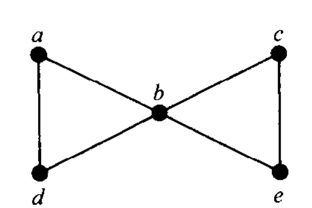


图 11.50　11.5 节练习 6 的图

(a)有欧拉回路和哈密顿回路.

(b)有哈密顿回路，但没有欧拉回路.

(c)有欧拉回路，但没有哈密顿回路.

(d)既没有欧拉回路，又没有哈密顿回路.

8. 给出下面图的例子：

(a)没有哈密顿链的图.

(b)没有哈密顿路径的有向图.

9. 假设 G 是图，它有 10 个顶点 a，b，c，d，e 和 x，y，u，v，w，前五个顶点的每一个顶点连接后五个顶点的每一个顶点.

(a)Ore 定理(定理 11.8)表明 G 有哈密顿回路吗？为什么？

663~664

(b)狄拉克定理(推论 11.8.1)表明 G 有哈密顿回路吗？为什么？

(c)G 有哈密顿回路吗？

10. (a)确定图 11.45 和图 11.46 中的每一图的闭包 $c(G)$.

(b)对于图 11.45 和图 11.46 中的哪些图，定理 11.9 表明存在哈密顿回路？

11. 给出一个图的例子，它的闭包没有哈密顿回路.

12. 证明：如果 G 有哈密顿回路，那么 $c(G)$ 不一定是完全的.

13. 对于图 11.47 和 11.48 中的哪些有向图，定理 11.10 表明存在哈密顿循环？

14. 对于图 11.47 和 11.48 中的哪些有向图，定理 11.11 表明存在哈密顿循环？

15. 一个图是*度 k 规则的*(regular of degree k)，如果每一个顶点都有同样的度数 k. 证明 G 有哈密顿回路，如果

(a)G 有 11 个顶点，且是度 6 规则的.

(b)G 有 13 个顶点，且是度 6 规则的.

16. 假设 $n \geqslant 4$，有 n 个人，其中任意两个人合起来都认识其余的 $n-2$ 个人. 证明可以把这 n 个人安排就座于一个圆桌，使得每个人都就坐于两个朋友之间.

17. 假设通过不断地增加连接度数和至少为 n 的不邻接顶点的边，最终我们可以得到图 H，而且不能再继续进行下去. 假设通过用不同的顺序进行这一构造过程，最终我们得到图 K，而且不能再继续下去. 证明 $H=K$[*提示*：如果 $H \neq K$，寻找在第一次构造过程中被加入到 G 中的第一个不是 K 的边的边].

18. 证明等式(11.15)表明 D 是强连通的.

19. 假设定理 11.10 的假定，当等式(11.13)中的 n 被替换成 $n-1$ 时成立. 使用定理 11.10 证明 D 有哈密顿路径.

20. 证明：推论 11.8.1 为假，如果“每个顶点的度数至少为 $n/2$”被替换成“每个顶点的度数至少为 $(n-1)/2$”.

21. 假设例 11.5 的重新设置成本是以下面的矩阵 C 的形式给出的：

$$C=\begin{bmatrix} 0 & 0 & 0 & 6 & 5 & 5 & 3 & 8 & 4 & 9 \\ 6 & 0 & 1 & 7 & 3 & 9 & 8 & 0 & 6 & 8 \\ 11 & 4 & 0 & 8 & 0 & 7 & 10 & 5 & 3 & 5 \\ 8 & 8 & 0 & 0 & 11 & 8 & 7 & 3 & 9 & 8 \\ 4 & 4 & 7 & 4 & 0 & 0 & 0 & 11 & 10 & 8 \\ 6 & 5 & 6 & 2 & 0 & 0 & 8 & 8 & 2 & 5 \\ 6 & 7 & 6 & 5 & 3 & 11 & 0 & 9 & 0 & 15 \\ 4 & 3 & 9 & 8 & 8 & 7 & 0 & 0 & 3 & 5 \\ 0 & 3 & 4 & 11 & 10 & 6 & 8 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 7 & 6 & 5 & 3 & 1 & 0 \end{bmatrix},$$

其中 c_{ij} 是从加工产品 i 的生产设备到加工产品 j 的生产设备的重新设置时的成本. 是否存在一种生产调度，在这一调度中没有重新设置成本发生？

665

22. (Pósa[1962])使用定理 11.9 证明：如果 G 是有至少三个顶点的图，且如果对于每一个 $1 \leqslant j \leqslant n/2$ 的整

数 j，度数不超过 j 的顶点的数量小于 j，那么 G 有哈密顿回路.

23. 证明或推翻下面关于 Ore 定理(定理 11.8)的变形. 假设 G 是有 n 个顶点的图，且 $n\geqslant 3$，并有下面的不等式成立：

$$\frac{\sum(\deg_G(x)+\deg_G(y))}{q}\geqslant n,$$

其中，分子中的和是对于所有不邻接顶点对 x 和 y 求和，而 q 等于这样的 x，y 对的数量；那么 G 是哈密顿图.

11.6　哈密顿链和哈密顿路径的应用

本节给出哈密顿链和哈密顿路径概念的若干应用. 11.6.2 节和 11.6.3 节在某些方面依赖于 11.6.1 节的内容. 其余各节是独立的，可以按任意顺序阅读. 特别地，可以快速浏览 11.6.1 节，然后阅读 11.6.3 节，或者只包含 11.6.4 节. 11.6.5 节可以作为 11.4.4 节的一个自然的后续来阅读.

11.6.1　锦标赛

设 (V, A) 是一个有向图，并假设对于 V 中所有的 $u\neq v$，(u, v) 在 A 中或 (v, u) 在 A 中，但二者不都在 A 中. 这样的有向图被称为*锦标赛*(tournament). 锦标赛出现于各种不同的场合. 明显的例子包括网球赛、篮球赛等的循环赛[⊖]. 在循环赛中，每一对赛手(参赛对)比赛，且每一对赛手中有一个且仅有一个成员可以打败另一个(我们假设不允许平局). 锦标赛也出现在*成对比较实验*(pair comparison experiment)中，在这样的实验中，我们从一个集合中选出每一对选项，要求受试验者或决策者告知喜欢其中哪一个，或哪一个更重要，哪一个更符合条件等等(读者也许要考虑第 4 章例 4.1 和其他关于优先选择的讨论来进行比较). 自然界中也发生锦标赛，其中给定物种中的特定个体统治相同物种的其他个体. 在这样的情况下，对于这个物种的每一对动物，有一个且仅有一个个体统治另一个个体. 这种统治关系在有关个体之间定义了*啄序*(pecking order)关系.

有时候，我们想要对锦标赛的选手进行*分级*(rank). 例如，我们要颁发第二名、第三名等等. 又例如，"选手"是某个工作的候选人，而锦标赛表示候选人中的优先选择. 这时，我们的首选工作候选人也许不接受这个工作，那么我们需要提前确定第二选择、第三选择等等. 寻找选手分级的一个方法就是观察每一个锦标赛是否有哈密顿路径. 这一观察结果表明(我们将做简短证明)，我们可以把 n 个选手标记为 u_1，u_2，…，u_n，使得 u_1 打败 u_2，u_2 打败 u_3，…，u_{n-1} 打败 u_n. 这样的标签给出选手的分级：u_1 被排在第一位，u_2 是第二位，以此类推. 为了说明，考虑图 11.51 的锦标赛. 这里有一条哈密顿路径，因此，给出的分级是：Jones，Smith，Green，Brown，White，Black. 寻找选手分级的另一种方法是，使用练习 20 所定义的积分序列(参见练习 23 和练习 28).

666

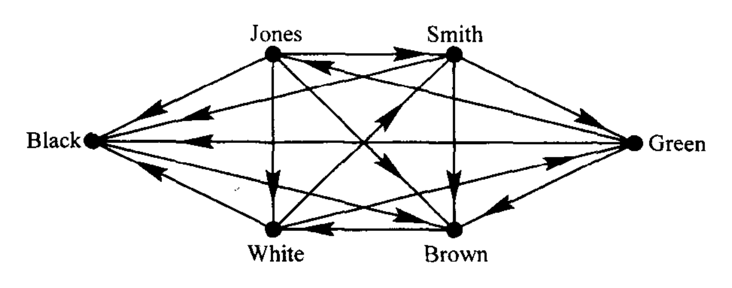


图 11.51　循环乒乓球锦标赛

定理 11.12(Redei[1934])　*每一个锦标赛 (V, A) 有哈密顿路径.*

证明[⊜]　我们通过对顶点数 n 使用归纳法进行证明. 如果 $n=2$，结果是显然的. 假设对于 n

⊖　读者应该把循环赛和更加普通的淘汰赛区分开来.

⊜　本证明可以跳过.

个顶点的锦标赛结果成立，现在考虑有 $n+1$ 个顶点的锦标赛(V, A). 设 u 是任意一个顶点，考虑由 $V-\{u\}$生成的子图. 容易证明这个子图仍然是一个锦标赛. 因此，根据归纳法假设，它有哈密顿路径 $u_1, u_2, \cdots, u_n$. 如果存在一条从 u 到 u_1 的弧，那么 $u, u_1, u_2, \cdots, u_n$ 是(V, A)的哈密顿路径. 如果不存在从 u 到 u_1 的弧，那么，因为(V, A)是一个锦标赛，存在一条从 u_1 到 u 的弧. 设 i 是存在从u_i 到 u 的弧的最大整数. 如果 i 等于 n，那么存在从 u_n 到 u 的弧，于是我们得出结论 $u_1, u_2, \cdots, u_n, u$ 是(V, A)的哈密顿路径. 如果 $i<n$，则存在从 u_i 到 u 的弧，根据 i 的定义，不存在从 u_{i+1}到 u 的弧. 因为(V, A)是锦标赛，所以存在从 u 到 u_{i+1}的弧. 因此，$u_1, u_2, \cdots, u_i, u, u_{i+1}, \cdots, u_n$ 是(V, A)的哈密顿路径. ■

如果 $u_1, u_2, \cdots, u_n$ 是锦标赛(V, A)中的一条哈密顿路径，那么正如我们已观察的那样，可以使用这条哈密顿路径来定义选手的分级. 遗憾的是，这个锦标赛中可能存在其他的哈密顿路径. 在我们的例子中，Smith，Brown，White，Green，Jones，Black 就是另一个哈密顿路径. 这时，存在多种可能的选手分级. 情况可能更糟. 在图 11.52 的锦标赛中，事实上，对于每一个顶点 a 和每一个可能的分级 r，存在哈密顿路径，对于这个哈密顿路径，a 有分级 r.

667

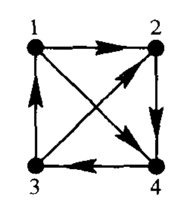


图 11.52　对每一个顶点 a 和分级 r，存在哈密顿路径，在这一路径下 a 有分级 r

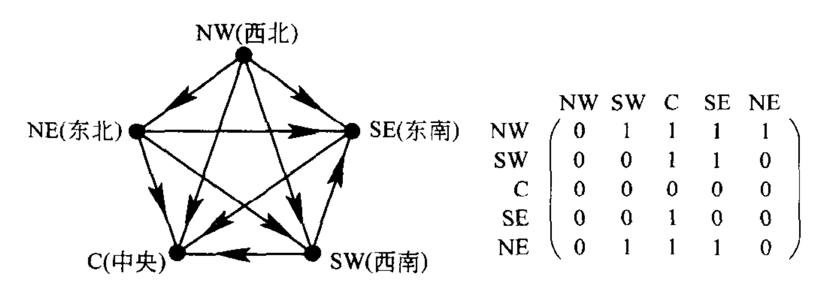
一般地，给定可能分级的集合，我们可能希望尝试选择一个"合意"分级. 在可供选择的可能的分级中寻找一个合意分级的问题，是一个相当困难的问题，这一问题的讨论超出了本书的范围(参见 Roberts[1976, Ch. 7]).

下面我们要问是否存在这样的环境，在该环境下不会出现有多种不同分级的问题. 即是否存在锦标赛有唯一哈密顿路径的环境？我们如下回答这一问题：一个锦标赛是*传递的*(transitive)，如果只要(u, v)是一条弧，而且(v, w)也是一条弧，那么(u, w)就是一条弧[⊖]. 图 11.51 的锦标赛不是传递的，因为 Jones 打败 Smith，而 Smith 打败 Green，但是 Jones 没能打败 Green.

定理 11.13　*一个锦标赛有唯一哈密顿路径，当且仅当这个锦标赛是传递的.*

证明　参见练习 27. ■

现在，我们考虑把定理 11.13 运用于在选择方案中决策者的优先选择，或者他的选择方案相对重要程度的分级等等. 通过优先选择的成对数据比较就可以得到结果：例如，如图 11.53 所示. 这里，传递性成立，所以存在唯一的哈密顿路径. 这一路径是西北、东北、西南、东南、中央.



	NW	SW	C	SE	NE
NW	0	1	1	1	1
SW	0	0	1	1	0
C	0	0	0	0	0
SE	0	0	1	0	0
NE	0	1	1	1	0

图 11.53　地理区域间的优先选择中的成对比较实验的结果. 矩阵中的 x，y 项是 1 当且仅当选择 x 而不是 y. 在所对应的有向图中，从 x 到 y 的弧表示优先选择 x 而不是 y

⊖ 传递性在第 4 章起着重要的作用. 有时候，对于有向图传递性的定义是不同的：仅当 $u\neq w$ 时，要求弧(u, v)和(v, w)的存在蕴涵弧(u, w)的存在. 对于锦标赛，如果(u, v)和(v, u)都是弧，那么条件 $u\neq w$ 自动成立. 为什么？

在优先选择的研究中，通常假设(或要求)决策者的优先选择定义一个传递锦标赛，即如果他较之 v 选择 u，较之 w 选择 v，那么他较之 w 要选择 u. 可以证明，这等价于假设决策者可以唯一地分级他表达优先选择的选择方案. 读者也许要与例 4.1 中的优先选择的讨论以及与 4.2 节中的各种顺序关系的讨论加以比较. 以 4.2 节的术语来说，锦标赛是一个严格线性序(见练习 16).

668

实施成对比较实验进而得到优先选择可能是一件乏味的工作. 如果存在 n 种选择，则需要进行 $\binom{n}{2}=n(n-1)/2$ 次比较，即使对于适当的 n，这个比较次数也可能很大. 然而，如果我们相信这个对象是传递的，那么我们知道成对比较实验相当于一个传递锦标赛，因此相当于选择方案的唯一分级. 因此，我们是在尝试着排序 n 个项的集合. 恢复这一分级实际需要多少次比较呢? 由此，恢复原来的锦标赛需要多少次比较呢? 这一问题是我们已在 3.6 节中讨论过的排序问题. 在那里，我们已指出好的排序算法大约需要 $n\log_2 n$ 次比较，与 $n(n-1)$ 相比这是一个更小的数字.

11.6.2　拓扑排序

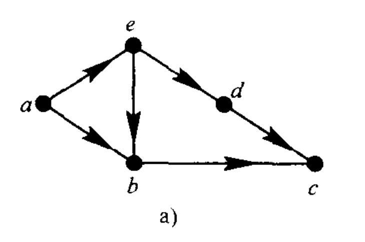
可以证明，一个锦标赛是传递的当且仅当它是*非循环的*(acyclic)，即没有循环(练习 26). 在一个传递的或非循环的锦标赛中寻找唯一哈密顿路径是如下更一般问题的一种特殊情况. 假设已知有 n 顶点的有向图 D. 使用整数 1，2，…，n 标记 D 的顶点，使得 D 的每一条弧都从较小标签的顶点指向较大标签的顶点. 这样的标签称为 D 的*拓扑顺序*(topological order)，而寻找这样一个顺序的过程称为*拓扑排序*(topological sorting).

定理 11.14　*有向图 D 有拓扑顺序当且仅当 D 是非循环的.*

证明　显然，如果存在拓扑顺序 1，2，…，n，那么 D 不能有循环. 反之，假设 D 是非循环的. 根据练习 31，存在没有进入弧的顶点 x_1. 标记 x_1 为 1，并从 D 中删除这个顶点. 此时，得到的有向图仍然是非循环的，因此有没有进入弧的顶点 x_2. 标记 x_2 为 2；以此类推. 这显然给出一个拓扑顺序. ■

669

通过寻找如图 11.54a 所示的有向图 D 的一个拓扑顺序，来说明定理 11.14 证明中的构造法. 首先，选择顶点 a. 删除 a 之后，我们选择顶点 e. 删除 e 之后，我们要在 b 和 d 之间做一个选择，比如说我们选择 b. 那么删除 b 之后，我们选择 d. 最后我们留下 c. 这给出如图 11.54b 所示的标签. 所描述的这一过程存在的一个问题是在每一步，我们必须为了选择下一个顶点而搜索整个剩余的有向图. 关于基于 11.1 节所描述的深度优先搜索过程的更高效算法，请参见 Cormen，Leiserson 和 Rivest[1999]、Golumbic[1980]、Reingold，Nievergelt 和 Deo[1977].



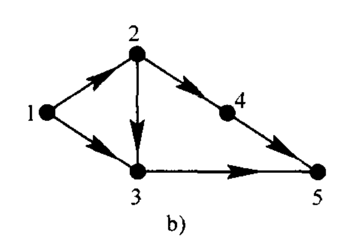


图 11.54　b 给出 a 的有向图的一个拓扑顺序

拓扑排序有着广泛的应用. 例如，它出现于工程规划(Deo[1974])的*活动网络*(activity network)分析中. 把一个大工程分解成称为*活动*(activity)的若干任务. 某些活动必须在其他活动开始之前完成，例如一个产品必须在刷油漆之前打磨. 如下构建有向图 D. 顶点是活动，如果 x 必须超前于 y，则存在从活动 x 到活动 y 的弧. 我们设法寻找一个完成活动的顺序，使得如果在 D 中存在从 x 到 y 的一条弧，那么 x 在 y 之前出现. 这要求 D 的拓扑排序. 当且仅当 D 没有循环时，可以做到这一点. 例如，当我们排列术语表中的字并保证没有字在被定义之前使用时，就会出现类似的情况. 这也需要拓扑排序.

11.6.3 运筹学中的调度问题㊀

运筹学中的很多调度问题，如 11.6.2 节所讨论的活动网络等，都涉及寻找执行特定数量操作的顺序. 我们通常要寻找一个最优顺序. 有时候，这样的问题可以通过寻找哈密顿路径得以解决. 我们已经在例 11.5 中看到了这方面的例子. 这里，我们给出另一个例子，相关的问题将在 13.2.3 节讨论.

假设一名印刷业者有 n 本不同的书需要印刷. 他有两台机器，一台印刷机和一台装订机. 一本书必须在装订之前印刷. 装订机操作工人比印刷机操作工人赚的钱要多，而且这些收入必须在装订机第一次启动到所有书装订完毕这段时间内支付. 这些书以什么样的顺序印刷才能使得装订机操作工人的总支付达到最小?

670

设 p_k 是印刷第 k 本书所需的时间，而 b_k 是装订第 k 本书所需的时间. 让我们做下面的特殊假设，对于所有的 i 和 j，或者 $p_i \leqslant b_j$，或者 $p_j \leqslant b_i$. 注意，现在有可能寻找书的排列顺序，使得如果书以这一顺序印刷和装订，在第一本书印刷完成之后装订机保持忙碌而没有空闲时间. 这显然最小化支付给装订操作工人的钱. 为了寻找到这一理想顺序，画出一个有向图，它有从 i 到 j 的弧当且仅当 $b_i \geqslant p_j$. 这时，这个有向图包含一个锦标赛，所以根据定理 11.12，这个图存在一条哈密顿路径. 这个路径给出一个要求的顺序. 这一问题的更一般且没有特殊假设的情况下的论述，请参见 Johnson[1954](还可参考练习 38).

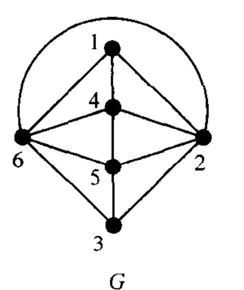
11.6.4 设备设计㊁

图论在诸如工厂、医院、学校、高尔夫球场等等的物理设施的设计中有着广泛的应用. 在这样的设计问题中，我们有若干个需要定位的区域，并希望使某些区域相互接壤. 我们画出一个图 G，即*关系图*(relationship graph)，它的顶点是所讨论的区域，而且当两个区域应该相互接壤时，其间有一条边. 如果 G 是平面图，那么这就形成一个地图(参见例 3.19)，而所对应的地图表示一个设计：其中两个被 G 中的边连接的区域(国家)事实上共享一面墙(边界). 图 11.55 给出一个平面图和相应的设施布局.

如果关系图 G 不是平面图，那么我们设法消除某些需求，即删除 G 的某些边，设法寻找 G 的平面支撑子图 G'.

671

这时，可以使用 G' 去构建设计. 下面是一个寻找用于设施规划的 G' 的方法(源于 Demourcron，Malgrance 和 Pertuiset[1964]). 确定 G 是否有哈密顿回路. 如果有哈密顿回路，寻找这样的一个回路 $C=u_1, u_2, \cdots, u_n$. 按照这一顺序把各顶点定位在一个圆上. 例如，考虑图 11.56 的图 G. 那么 $C=1, 2, 3, 4, 5$ 是一条哈密顿回路，它被表示成图 11.56 中的圆上的点. 如下构建一个新图 H. H 的顶点是 G 中的那些不是 C 的边的边. H 的两条边邻接当且仅当把所对应的弦画在由 C 所确定的圆内时，这些弦相交. 对于我们例子中的弦，参见图 11.56，对于图 H，参见图 11.57. 假设 H 是 2 可着色的，比如说使用红和蓝着色. 那么 G 是平面图. 为了弄明白这一点，只需注意所有红色的弦都可画在由 C 所确定的圆内，而所有蓝色的弦都可画在由 C 确定的圆外，没有边相交. 例如，在图 11.57 的图 H 中，我们可以着色顶点 {1，3} 和 {1，4} 为红色，而着色顶点 {2，4} 和 {2，5} 为蓝色. 这时，我们得到如



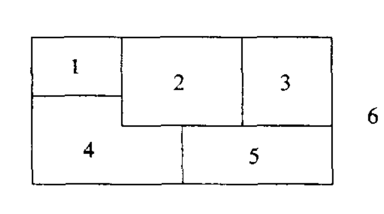


图 11.55 关系图 G 和对应的设计

㊀ 本小节基于 Berge[1962]、Johnson[1954]和 Liu[1972].

㊁ 本小节基于 Charchra，Ghare 和 Moore[1979].

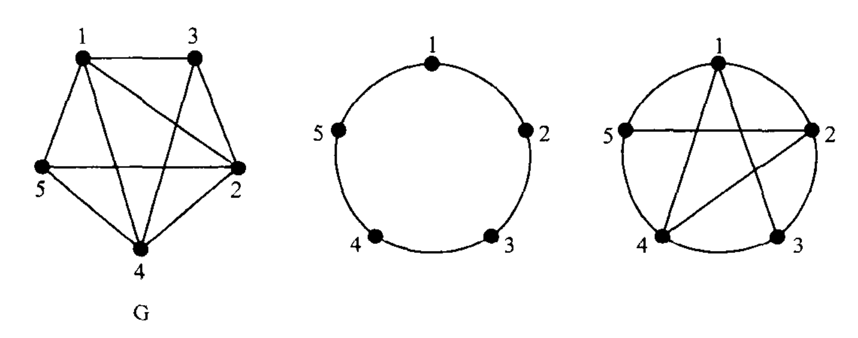


图 11.56　关系图 G、分布在圆上的哈密顿回路的顶点、G 的剩余边表示成这个圆的弦

图 11.58 所示的 G 的平面图. 如果 H 不是 2 可着色的，那么我们寻找 2 可着色的 H 的最大子图 K. 把 K 的边加入到 C，得到一个 G 的支撑子图的平面图 G'. 利用 G' 确定设计. 例如，假设 G 是通过加入边{3, 5}，从图 11.56中的图 G 而得到的. 那么，利用相同的哈密顿回路 C，我们看到 H 是如图 11.59 所示的图. 这个 H 不是 2 可着色的. 我们必须消除某个顶点来得到 K，例如消除顶点{3, 5}. 所得到的图 G' 是图 11.56所示的图 G，它是一个平面图.

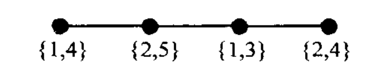


图 11.57　从图 11.56 得到的图 H

672

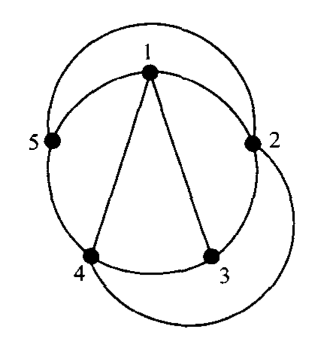


图 11.58　图 11.56 中的图 G 的平面图

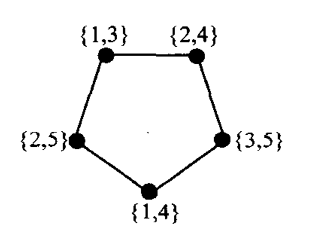


图 11.59　如果把边{3, 5}加入到图 11.56 中的 G，得到图 H

11.6.5　杂交顺序[⊖]

在例 11.2 中，我们描述了根据所有 k 长度子串的列表 L 确定 DNA 串 S 的杂交顺序问题. 这一问题的另一个解决方法是构建一个有向图 $E(L)$，这个有向图的顶点是 L 中的串，而且如果 $a_2=b_1$，$a_3=b_2$，…，$a_k=b_{k-1}$，那么它有一条从 L 中的 $a_1a_2\cdots a_k$ 到 L 中的 $b_1b_2\cdots b_k$ 的弧. 对应于(11.2)列表 L 的有向图 $E(L)$ 如图 11.60 所示. 如在例 11.2 中那样，假设每个长度为 k 的子串在 L 中出现不超过一次，那么不难看出，在访问 $E(L)$ 的每一个顶点正好一次的路径与 k 长度子串对应于 L 中的串的 DNA 串之间存在着一一对应. 我们使用这样的路径中的每一个顶点的第

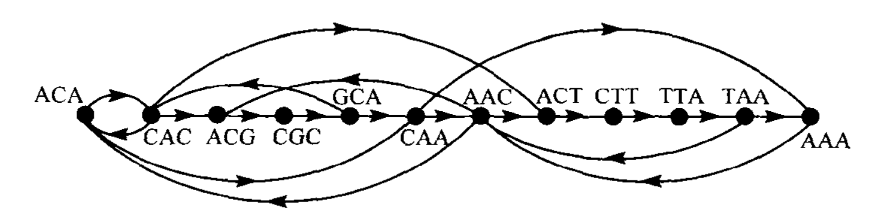


图 11.60　从(11.2)的列表 L 得到的有向图 $E(L)$

⊖ 本小节基于 Pevzner[2000].

一个字母，并以路径上最后顶点的后 $k-1$ 个字母为结尾. 因此，由 L 所生成的这个 DNA 串对应于 $E(L)$ 中的哈密顿路径. 在图 11.60 中，哈密顿路径 ACA，CAC，ACG，CGC，GCA，CAA，AAC，ACT，CTT，TTA，TAA，AAA 对应于 DNA 串 ACACGCAACTTAAA. 然而，还存在其他哈密顿路径. 例如，路径 ACG，CGC，GCA，CAA，AAA，AAC，ACA，CAC，ACT，CTT，TTA，TAA. 因为寻找哈密顿路径的问题是 NP 完全问题，所以寻找未知 DNA 片段的方法不如在例 11.2 中所描述的欧拉路径方法有效.

673

本节练习

1. 对于下面的每一种情况，确定对应于二元关系的有向图是否是锦标赛.
   (a)二元关系(X, R)，其中 $X=\{1, 2, 3, 4\}$，R 由等式(4.1)定义.
   (b)二元关系(X, S)，其中 $X=\{1, 2, 3, 4\}$，S 由等式(4.2)定义.
2. 对于 4.1 节练习 1 中的每一个二元关系，确定所对应的有向图是否是锦标赛.
3. 对于练习 1 中的每一个二关系，确定对应于这一二元关系的有向图是否有拓扑顺序，如果存在这样的顺序，寻找一个这样的顺序.
4. 对于 4.1 节练习 1 中的每一个二元关系，确定对应于这一二元关系的有向图是否有拓扑顺序.
5. 对于表 4.2 定义的二元关系的每一个性质，确定它对于锦标赛是否(i)总成立，(ii)有时候成立，(iii)不成立.
6. 证明传递锦标赛是 4.2 节定义的严格线性序.
7. 对于图 11.61 中的每一个锦标赛，寻找所有的哈密顿路径.
8. 对于根据表 11.3 的优先选择数据定义的锦标赛，寻找它的所有哈密顿路径.

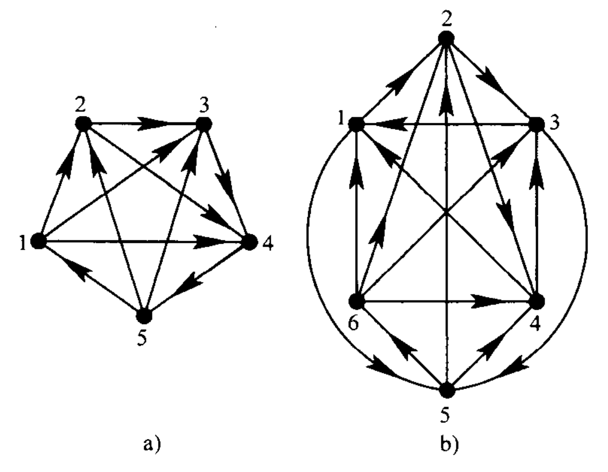


图 11.61 11.6 节练习的锦标赛

表 11.3 作为生活地点的各城市间优先选择的一个成对比较实验结果
(项 i，j 等于 1，当且仅当优先选择 i 而不选择 j)

	纽约	波士顿	旧金山	洛杉矶	休斯顿
纽约	0	0	0	0	1
波士顿	1	0	0	1	1
旧金山	1	1	0	1	1
洛杉矶	1	0	0	0	0
休斯顿	0	0	0	1	0

9. 有 4 个或更少个顶点的锦标赛中，哪些是传递的?
10. 画出一个有 5 个选手的传递锦标赛的有向图.
11. 寻找图 11.62 中每一个有向图的拓扑顺序.
12. 在 11.6.3 节的书籍印刷问题中，假设 $b_1=3$，$b_2=5$，$b_3=8$，$p_1=6$，$p_2=2$，$p_3=9$. 寻找一个制书的最优顺序.
13. 如果 $b_1=10$，$b_2=7$，$b_3=5$，$p_1=11$，$p_2=4$，$p_3=8$，重复练习 12.
14. 为图 11.46 中的平面图 a 和 b 设计规划.
15. 如果 G 是图 11.45 中的图 a，寻找一条哈密顿回路 C 并如 11.6.4 节那样构造 H. 如果存在的话，利用 H 确定 G 的平面图示. 否则寻找平面图 G'.
16. 如果 G 是图 11.45 中的图 b，重复练习 15.

674~675

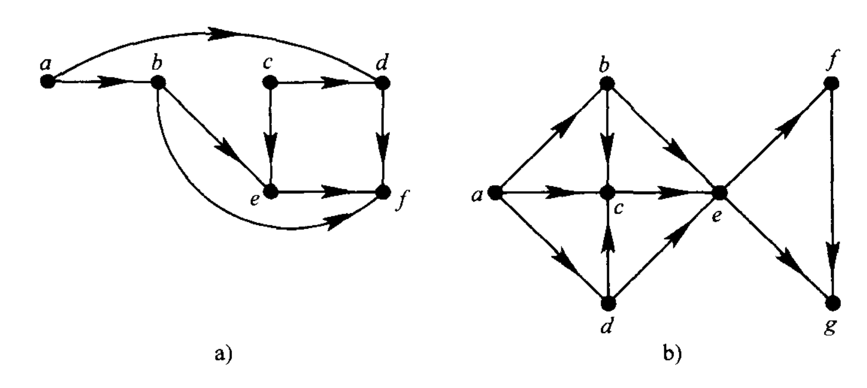


图 11.62 11.6 节练习 11 的有向图

17. 如果 G 是图 11.43 中的图 b，重复练习 15.

18. (a)对于列表 $L=\{$CCGA，CGCG，CGTG，CGAA，GCCG，GCGT，GTGC，GAAC，TGCC，AACC，ACCC$\}$，画出 11.6.5 节中的有向图 $E(L)$.

    (b)确定 $E(L)$ 中哈密顿路径的数量.

    (c)寻找 L 的相容 DNA 串.

19. (a)对于列表 $L=\{$AA，AC，AG，AT，CA，CT，GA，GT，TA，TC，TG，TT$\}$，画出 11.6.5 节中的有向图 $E(L)$.

    (b)$S=$ACTTCATGAGTAA 与 L 相容吗?

    (c)$S=$AATGAGTACTTCA 与 L 相容吗?

    (d)$S=$CTAGTACATGATA 与 L 相容吗?

20. 在一个锦标赛中，顶点 u 的*积分*(score)$s(u)$是 u 的出度数(这是 u 打败的选手数量). 如果顶点被标记为 1，2，…，n，且 $s(1)\leqslant s(2)\leqslant\cdots\leqslant s(n)$，序列$(s(1), s(2), \cdots, s(n))$称为这一锦标赛的*积分序列*(score sequence). 练习 20～练习 24 和练习 28 都是研究积分序列的. 寻找下面锦标赛的积分序列：

    (a)图 11.51 (b)图 11.53

21. 证明：如果$(s(1), s(2), \cdots, s(n))$是锦标赛的积分序列，那么有

$$\sum_{i=1}^{n} s(i) = \binom{n}{2}.$$

22. 下面的每一个序列可能是锦标赛的积分序列吗? 为什么?

    (a)(1，1，2，3) (b)(0，0，0，2，7) (c)(0，1，1，4，4)

    (d)(0，0，3，3) (e)(1，2，2，3，3，4)

23. 证明在锦标赛中，使用积分序列的选手分级可以不同于使用哈密顿路径的选手分级.

24. 使用积分序列画出下面锦标赛的有向图.

    (a)(0，1，2，3) (b)(2，2，2，2，2) (c)(1，1，1，4，4，4)

25. 证明一个锦标赛是传递的当且仅当它没有长度为 3 的循环.

26. 证明一个锦标赛是传递的当且仅当它是非循环的.

27. 如下证明定理 11.13.

    (a)证明：如果 D 是传递的，那么它有唯一哈密顿路径，方法是观察如果存在两条这样的路径，那么一定存在 u 和 v，且在一条路径中 u 跟在 v 的后面，而在另一条路径中 v 跟在 u 后面.

    (b)证明其逆叙，方法是观察在唯一的哈密顿路径 $u_1, u_2, \cdots, u_n$ 中，我们有$(u_i, u_j)\in A$ 当且仅当$i<j$.

676

28. (a)使用练习 27(b)的结果寻找有 n 个顶点的传递锦标赛的积分序列.

(b)证明对于一个传递锦标赛，由积分序列而得到的分级与由唯一哈密顿路径而得到的分级相同.

29. 假设一个锦标赛的顶点可以列表为 u_1，u_2，…，u_n 使得积分 $s(u_i)=n-i$. 这一竞赛有哈密顿路径吗?(给出证明或反例).
30. 使用练习 28(a)的结果，确定表 11.3 的优先选择数据是否是传递的.
31. 证明每一个非循环有向图都有没有入弧的顶点.
32. 在锦标赛中，顶点的三元组{x，y，z}称为**传递的**(transitive)，如果由 x，y 和 z 生成的子图是传递的，等价地，如果三个顶点中的一个打败其余两个. 这个顶点称为这个传递三元组的**传递者**(transmitter).

    (a)如何确定给定顶点 u 为传递者的传递三元组的数量?

    (b)证明：在锦标赛中，如果 $s(x)$是 x 的积分，那么有 $\sum_x \binom{s(x)}{x}$ 个传递三元组.

    (c)证明至少有四个顶点的每一个锦标赛必定有传递三元组.
33. 如果你还没有完成 3.6 节练习 40，那么现在完成它.
34. (Harary，Norman 和 Cartwright[1965])如果 D 有层次分配(3.6 节练习 40)，而 r 是 D 的最长简单路径的长度，证明 $r+1$ 是 D 的层次分配中不同层次(值 L_i)的最小数量.
35. 如果你还没有完成 3.6 节练习 41，那么完成它.
36. 在传递锦标赛中，如果顶点 u 有最大积分，那么 u 打败其他每一名选手.

    (a)证明在任意锦标赛中，如果 u 有最大积分，那么对于其他每一名选手 v，或者 u 打败 v，或者 u 打败一名打败 v 的选手(这一结果是由 Landau[1955]年发现的，当时他在研究小鸡间的啄序. 关于这个结果的扩展，参见 Maurer[1980]).

    (b)证明(a)中 u 为胜者的必要条件不是充分的.
37. (Camion[1959]和 Foulkes[1960])证明每一个强连通锦标赛有哈密顿循环[提示：证明对于 $k=3$，4，…，n，存在长度为 k 的循环，其中 n 是顶点的数量].
38. 11.6.3 节的制书问题中的另一个优化标准是，在尽可能短的时间内完成所有书的印刷和装订. 证明即使在 11.6.3 节的特殊假设的情况下，可能存在不对应于 11.6.3 节中所构造的有向图中的哈密顿路径的最优解.

677

参考文献

ABELLO, J., BUCHSBAUM, A., and WESTBROOK, J., "A Functional Approach to Extremal Graph Algorithms," *Algorithmica, 32* (2002), 437–458.

ABELLO, J., PARDALOS, P. M., and RESENDE, M. G. C., "On Maximum Clique Problems in Very Large Graphs," in J. Abello and J. Vitter (eds.), *External Memory Algorithms*, DIMACS Series, Vol. 50, American Mathematical Society, Providence, RI, 1999, 119–130.

ABELLO, J., and VITTER, J. (eds.), *External Memory Algorithms*, DIMACS Series, Vol. 50, American Mathematical Society, Providence, RI, 1999.

AHO, A. V., HOPCROFT, J. E., and ULLMAN, J. D., *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

AIELLO, W., CHUNG, F. R. K., and LU, L., "A Random Graph Model for Massive Graphs," *Proc. 32nd Annual ACM Symposium on Theory of Computing*, (2000), 171–180.

ATALLAH, M. J., "Parallel Strong Orientation of an Undirected Graph," *Inform. Process. Lett., 18* (1984), 37–39.

BAASE, S., *Computer Algorithms*, 2nd ed., Addison-Wesley Longman, Reading, MA, 1992.

BELTRAMI, E. J., *Models for Public Systems Analysis*, Academic Press, New York, 1977.

BERGE, C., *The Theory of Graphs and Its Applications*, Wiley, New York, 1962.

BERMOND, J.-C., BOND, J., MARTIN, C., PEKEČ, A., and ROBERTS, F. S., "Optimal

Orientations of Annular Networks," *J. Interconn. Networks, 1* (2000), 21–46.

BOESCH, F., and TINDELL, R., "Robbins' Theorem for Mixed Graphs," *Amer. Math. Monthly, 87* (1980), 716–719.

BONDY, J. A., and CHVÁTAL, V., "A Method in Graph Theory," *Discrete Math., 15* (1976), 111–136.

CAMION, P., "Chemins et Circuits Hamiltoniens des Graphes Complets," *C. R. Acad. Sci. Paris, 249* (1959), 2151–2152.

CHACHRA, V., GHARE, P. M., and MOORE, J. M., *Applications of Graph Theory Algorithms*, Elsevier North Holland, New York, 1979.

CHARTRAND, G., and LESNIAK, L., *Graphs and Digraphs*, 3rd ed., CRC Press, Boca Raton, 1996.

CHRISTOFIDES, N., *Graph Theory: An Algorithmic Approach*, Academic Press, New York, 1975.

CHUNG, F. R. K., GAREY, M. R., and TARJAN, R. E., "Strongly Connected Orientations of Mixed Multigraphs," *Networks, 15* (1985), 477–484.

CHVÁTAL, V., and THOMASSEN, C., "Distances in Orientations of Graphs," *J. Comb. Theory B, 24* (1978), 61–75.

CORMEN, T. H., LEISERSON, C. E., and RIVEST, R. L., *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1999.

DE BRUIJN, N. G., "A Combinatorial Problem," *Nedl. Akad. Wet., Proc., 49* (1946), 758–764; *Indag. Math., 8* (1946), 461–467.

DEMOURCRON, G., MALGRANCE, V., and PERTUISET, R., "Graphes Planaires: Reconnaissance et Construction de Representations Planaires Topologiques," *Recherche Operationelle, 30* (1964), 33.

DEO, N., *Graph Theory with Applications to Engineering and Computer Science*, Prentice Hall, Englewood Cliffs, NJ, 1974.

DIRAC, G. A., "Some Theorems on Abstract Graphs," *Proc. Lond. Math. Soc., 2* (1952), 69–81.

678

DRMANAC, R., LABAT, I., BRUKNER, I., and CRKVENJAKOV, R., "Sequencing of Megabase Plus DNA by Hybridization: Theory of the Method," *Genomics, 4* (1989), 114–128.

DROR, M., *Arc Routing: Theory, Solutions and Applications*, Kluwer, Boston, 2000.

EULER, L., "Solutio Problematis ad Geometriam Situs Pertinentis," *Comment. Acad. Sci. I. Petropolitanae, 8* (1736), 128–140. [Reprinted in *Opera Omnia*, Series 1–7 (1766), 1–10.]

EVEN, S., *Graph Algorithms*, Computer Science Press, Potomac, MD, 1979.

FOULKES, J. D., "Directed Graphs and Assembly Schedules," *Proc. Symp. Appl. Math, Amer. Math. Soc., 10* (1960), 281–289.

GHOUILA-HOURI, A., "Une Condition Suffisante D'existence d'un Circuit Hamiltonien," *C. R. Acad. Sci. Paris, 156* (1960), 495–497.

GOLOMB, S. W., *Shift Register Sequences*, Aegean Park Press, Laguna Hills, CA, 1982.

GOLUMBIC, M. C., *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

GOOD, I. J., "Normal Recurring Decimals," *J. Lond. Math. Soc., 21* (1946), 167–169.

GUSFIELD, D., *Algorithms on Strings, Trees and Sequences; Computer Science and Computational Biology*, Cambridge University Press, New York, 1997.

GUTIN, Z. G., "Minimizing and Maximizing the Diameter in Orientations of Graphs," *Graphs Combin., 10* (1994), 225–230.

HAN, X., "On the Optimal Strongly Connected Orientations of City Street Graphs: Over Twenty East-West Avenues or North-South Streets," mimeographed, Shanghai Computer Software Laboratory, Shanghai, China, 1989.

HARARY, F., NORMAN, R. Z., and CARTWRIGHT, D., *Structural Models: An Introduction to the Theory of Directed Graphs*, Wiley, New York, 1965.

HARARY, F., and PALMER, E. M., *Graphical Enumeration*, Academic Press, New York, 1973.

HELLY, W., *Urban Systems Models*, Academic Press, New York, 1975.

HOPCROFT, J. E., and TARJAN, R. E., "Algorithm 447: Efficient Algorithms for Graph Manipulation," *Commun. ACM, 16* (1973), 372–378.

HUTCHINSON, G., "Evaluation of Polymer Sequence Fragment Data Using Graph Theory," *Bull. Math. Biophys., 31* (1969), 541–562.

HUTCHINSON, J. P., and WILF, H. S., "On Eulerian Circuits and Words with Prescribed Adjacency Patterns," *J. Comb. Theory A, 18* (1975), 80–87.

JOHNSON, E. L., "Chinese Postman and Euler Tour Problems in Bi-Directed Graphs," in M. Dror (ed.), *Arc Routing: Theory, Solutions and Applications*, Kluwer, Boston, 2000, 171–196.

JOHNSON, S. M., "Optimal Two- and Three-Stage Production Schedules with Setup Times Included," *Naval Res. Logist. Quart., 1* (1954), 61–68.

KARP, R. M., and RAMACHANDRAN, V., "Parallel Algorithms for Shared-Memory Machines," in J. Van Leeuwen (ed.), *Handbook of Theoretical Computer Science*, Vol. A, Elsevier, Amsterdam, 1990, 869–941.

KOH, K. M., and TAN, B. P., "The Diameter of an Orientation of a Complete Multipartite Graph," *Discr. Math., 149* (1996), 331–356. (a) [See also addendum *Discr. Math., 173* (1997), 297–298.]

KOH, K. M., and TAN, B. P., "The Minimum Diameter of Orientations of a Complete Multipartite Graph," *Graphs Combin., 12* (1996), 333–339. (b)

679

KWAN, M. K., "Graphic Programming Using Odd or Even Points," *Chin. Math., 1* (1962), 273–277.

LANDAU, H. G., "On Dominance Relations and the Structure of Animal Societies III. The Condition for a Score Sequence," *Bull. Math. Biophys., 15* (1955), 143–148.

LAWLER, E. L., *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.

LIEBLING, T. M., *Graphentheorie in Planungs- und Tourenproblemen*, Lecture Notes in Operations Research and Mathematical Systems No. 21, Springer-Verlag, New York, 1970.

LIU, C. L., *Introduction to Combinatorial Mathematics*, McGraw-Hill, New York, 1968.

LIU, C. L., *Topics in Combinatorial Mathematics*, Mathematical Association of America, Washington, DC, 1972.

MAURER, S. B., "The King Chicken Theorems," *Math. Magazine, 53* (1980), 67–80.

MCCANNA, J. E., "Orientations of the n-Cube with Minimum Diameter," *Disc. Math., 68* (1988), 309–310.

MINIEKA, E., *Optimization Algorithms for Networks and Graphs*, Dekker, New York, 1978.

ORE, O., "Note on Hamilton Circuits," *Amer. Math. Monthly, 67* (1960), 55.

PEVZNER, P. A., "1-Tuple DNA Sequencing: Computer Analysis," *J. Biomol. Structure Dynamics, 7* (1989), 63–73.

PEVZNER, P. A., *Computational Molecular Biology*, MIT Press, Cambridge, MA, 2000.

PEVZNER, P. A., and LIPSHUTZ, R., "Towards DNA Sequencing Chips," in *Proc. 19th Symp. Math. Found. Comp. Sci.*, LNCS 684, Springer, New York, 1994, 143–158.

PLESNIK, J., "Remarks on Diameters of Orientations of Graphs," *Math. Univ. Comeniance, 46/47* (1986), 225–236.

PÓSA, L., "A Theorem Concerning Hamiltonian Lines," *Magyar Tud. Akad. Mat. Kutató Int. Közl., 7* (1962), 225–226.

RÉDEI, L., "Ein kombinatorischer Satz," *Acta Litt. Sci. (Sect. Sci. Math.), Szeged, 7* (1934), 39–43.

REINGOLD, E. M., NIEVERGELT, J., and DEO, N., *Combinatorial Algorithms: Theory and Practice*, Prentice Hall, Englewood Cliffs, NJ, 1977.

REINGOLD, E. M., and TARJAN, R. E., "On a Greedy Heuristic for Complete Matching," *SIAM J. Comput., 10* (1981), 676–681.

ROBBINS, H. E., "A Theorem on Graphs, with an Application to a Problem of Traffic Control," *Amer. Math. Monthly, 46* (1939), 281–283.

ROBERTS, F. S., *Discrete Mathematical Models, with Applications to Social, Biological, and Environmental Problems*, Prentice Hall, Englewood Cliffs, NJ, 1976.

ROBERTS, F. S., *Graph Theory and Its Applications to Problems of Society*, NSF-CBMS Monograph No. 29, SIAM, Philadelphia, 1978.

ROBERTS, F. S., and XU, Y., "On the Optimal Strongly Connected Orientations of City Street Graphs. I: Large Grids," *SIAM J. Discr. Math., 1* (1988), 199–222.

ROBERTS, F. S., and XU, Y., "On the Optimal Strongly Connected Orientations of City Street Graphs. II: Two East-West Avenues or North-South Streets," *Networks, 19* (1989), 221–233.

ROBERTS, F. S., and XU, Y., "On the Optimal Strongly Connected Orientations of City Street Graphs. III: Three East-West Avenues or North-South Streets," *Networks, 22* (1992), 109–143.

ROBERTS, F. S., and XU, Y., "On the Optimal Strongly Connected Orientations of City Street Graphs. IV: Four East-West Avenues or North-South Streets," *Discr. Appl. Math., 49* (1994), 331–356. 680

TARJAN, R. E., "Depth-First Search and Linear Graph Algorithms," *SIAM J. Comput., 1* (1972), 146–160.

TUCKER, A. C., and BODIN, L. D., "A Model for Municipal Street-Sweeping Operations," in W. F. Lucas, F. S. Roberts, and R. M. Thrall (eds.), *Discrete and System Models*, Vol. 3 of *Modules in Applied Mathematics*, Springer-Verlag, New York, 1983, 76–111.

VAN AARDENNE-EHRENFEST, T., and DE BRUIJN, N. G., "Circuits and Tress in Oriented Linear Graphs," *Simon Stevin, 28* (1951), 203–217.

VISHKIN, U., "On Efficient Parallel Strong Orientation," *Inform. Process. Lett., 20* (1985), 235–240.

WHITNEY, H., "Congruent Graphs and the Connectivity of Graphs," *Amer. J. Math., 54* (1932), 150–168.

WOODALL, D. R., "Sufficient Conditions for Circuits in Graphs," *Proc. Lond. Math. Soc., 24* (1972), 739–755. 681



第四部分　组合优化

第12章　匹配与覆盖

12.1　一些匹配问题

本章研究属于匹配问题和覆盖问题两个范畴中的若干问题. 我们从两个方面考查这些问题，首先考虑存在问题，然后考虑优化问题. 因此，本章是从组合数学的第二个基本问题到第三个基本问题的过渡，即存在问题到优化问题的过渡. 我们先给出若干例子.

【例 12.1 工作分配(再探例 5.10)】 在例 5.10 中，我们讨论了工作分配的问题. 一般地，可以如下阐述这一问题. 有 n 名工人和 m 项工作. 每名工人适合其中的某些工作. 给每一名工人分配一项工作，确保他适合这一项工作，并确保没有两名工人得到相同的工作. 在例 5.10 中，我们讨论了计数实现这种分配的方法数量. 我们使用车多项式完成这一计数. 这里，我们要问一个存在问题：是否存在一个分配，它给每名工人分配一项他所适合的工作，并确保没有两名工人做同一项工作？随后，我们问一个优化问题：最好的分配是什么？

从图论的角度阐述存在问题是很方便的. 如下构建一个图 G. G 有 $m+n$ 个顶点，每名工人对应一个顶点，每项工作对应于一个顶点. 用边把工人与他所适合的每一项工作连接起来. 不存在其他边. 图 12.1 给出一个具体的工作分配问题的结果图 G. 图 G 是*二部图*(bipartite graph)(X, Y, E)，这样的图的顶点集合被分成两个集合 X 和 Y，且有边集合 E，使得 E 中所有边都在集合 X 和 Y 之间(参见 3.3.4 节).

我们寻找的这种工作分配可以通过下面的方法表示：如果工人 x 被分配做工作 y，则用波形边取代 x 到 y 的边. 图 12.2 给出这样的一个分配. 注意，在这一分配中，每一个顶点至多在一条波形边上. 这对应于每名工人至多被分配给一项工作，而每项工作至多分配给一名工人. 图 G 中的边的集合 M 称为*匹配*(matching)，如果 G 中的每个顶点在 M 的至多一条边上. 因此，我们要寻找一个匹配. 在匹配 M 中，一个顶点称为是*饱和的*(saturated)，如果它在 M 的某条边上. 我们寻找使对应于工人的每一顶点饱和的匹配. 我们首先研究二部图的匹配，然后再研究任意图的匹配. 图 12.3 给出一个非二部图的匹配. ■

683

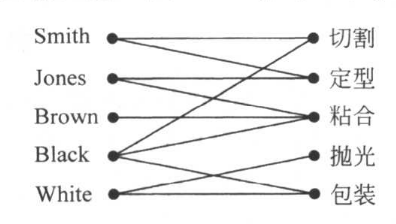


图 12.1　从工人 x 到工作 y 的边表示 x 适合工作 y

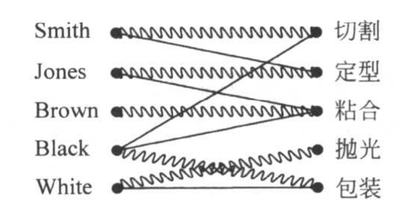
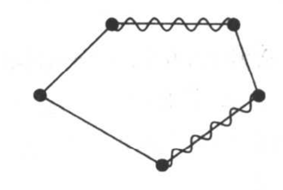


图 12.2　使用波线边表示的图 12.1 的工作分配图


图 12.3　非二部图中的匹配 M

【例 12.2 存储计算机程序(再探例 5.10)】 在例 5.10 中，我们还讨论了给计算机程序分配存储位置的问题. 一般地阐述这一问题，我们可以认为 n 个程序和 m 存储位置构成一个图的顶点. 如果位置 y 对于程序 x 有充分的存储容量，那么我们在 x 和 y 之间放置边. 我们寻找程序到存储位置的分配，使得每一个存储位置至多得到一个程序，而且每一个程序正好被分配到一个存储位置，这个位置对这个程序有充分的存储容量. 因此，我们在对应的二部图中寻找一个匹配，这个匹配使对应于程序的所有顶点饱和. 与例 5.10 情况对照，在那里，我们感兴趣的是计数这样的分配数量，而这里，我们感兴趣的问题是是否存在这样的分配. ■

【例 12.3 天花疫苗接种】 在制定可能爆发天花的计划中，地方卫生局设法给居住在某城市 684 的每一个市民分配一个接种诊所. 他们要寻找一种分配，使得每一个人分配到一个离他的家不超过 10 英里的诊所. 我们可以考虑一个二部图，其顶点集合 X 是居民，而顶点集合 Y 是诊所. 一个居民与他的家 10 英里内的任意诊所之间由一条边相连接. 注意，这不是一个匹配问题，因为我们允许多个人被分配到一家. 然而，如果我们认识到每一个诊所有接种的最大容量，并用由所有诊所的可能预约时间组成的集合取代 Y，那么我们可以重新阐述它为一个匹配问题. 这时，从一个居民到距他的家 10 英里内的所有诊所的所有预约时间有一条边，且我们寻找一个使 X 的每个顶点饱和的匹配. ■

【例 12.4 英国皇家空军的飞行员(Berge[1973], Minieka[1978])】 第二次世界大战中的英国战役期间，英国皇家空军(RAF)有许多来自外国的飞行员. RAF 必须给每一架飞机分配两名飞行员，而且总是想给同一架飞机分配两名语言一致且训练相似的飞行员. 我们可以如下翻译 RAF 问题. 给定可用于执行某项任务的飞行员集合，把他们当作一个图的顶点，当且仅当两名飞行员可以一起飞行时，将两个顶点，用一条边连接. 那么 RAF 想要这个图中的一个匹配(它没有必要是二部图). 另外，他们关心的是尽可能多驾驶飞机. 因此，他们感兴趣于*最大基数匹配*(maxiimum-cardinality matching)，具有最多可能边数的图匹配. 在 12.4 节和 12.5 节中，我们将解决要求最大基数匹配的组合优化问题. ■

【例 12.5 房地产交易(Minieka[1978])】 房地产代理商在给定时间有一个潜在买主的集合 X 以及要出售的房子的集合 Y. 如果买主 x 购买房子 y，那么设 r_{xy} 是代理商的收益. 纯粹从金钱的角度看，代理商想要使买主和房子匹配，以便最大化相应的 r_{xy} 的和. 我们可以通过设顶点 X 和 Y 定义一个*完全二部图* $G=(X, Y, E)$ 来表示这个代理商问题；即我们取 X 和 Y 之间所有可能的边. 这时，我们寻找 G 中的一个匹配，这个匹配使得权的和最大. 更一般地，我们可以考虑下面的*最大权匹配问题*(maximum-weight matching problem). 假设 G 是任意的图(不必是二部图)，且在每一条边 $\{x, y\}$ 上有权(实数) r_{xy}. 如果 M 是 G 的一个匹配，那么我们定义

$$r(M)=\sum\{r_{xy}: \{x, y\}\in M\}.$$

我们寻找 G 的*最大权匹配*(maximum-weight matching)，这是使得对于所有 G 的匹配 M'，$r(M)\geq r(M')$ 成立的匹配 M. ■

【例 12.6 最优分配问题】 让我们返回到例 12.1 的工作分配问题，但是增加每名工人适合每 685 项工作的简化假设. 这时，“适配”图 G 是一个完全二部图. 我们再假设，真对工人 x 对工作 y 的可能能力(输出)指定给他一个等级 r_{xy}. 我们寻找工人到工作的一个分配，这一分配给每名工人一项工作，每一项工作的人数不超过一人，而且这一分配使得等级之和最大. 寻找这样的分配的问题称为*最优分配问题*(optimal assignment problem). 如果工人的数量与工作的数量至少相同，那么这一问题是可解的. 它要求的是一个完全二部图 G 的一个匹配，这个匹配使工人的集

合饱和，而且至少与其他使工人集合饱和的任意匹配有相同大小的权和. 但是，显然每一个最大权匹配都使工人的集合饱和. 因此，最优分配问题可以简化成一个完全二部图的最大权匹配问题. 我们将在 12.7 节中返回到这一最优分配问题，在那里，我们给出解决这一问题的算法.（通过 12.4 节的内容，特别是推论 12.5.1，可以理解这一算法.） ■

【例 12.7 天花疫苗接种(再探例 12.3)】 在例 12.3 中，我们试图给每一个人指定距离他的家 10 英里以内的接种诊所. 现在，假设我们希望给每个人指定最近的可能接种诊所. 我们可以设 r_{xy} 表示人 x 到预约时间用 y 表示的接种诊所的距离. 那么，我们寻找可以最小化所有 x，y 的 r_{xy} 的和的分配，其中 x 被指定给 y. 这是例 12.6 的优化分配问题的最小化的变体. 如果人在不同诊所的不同预约时间有优先选择的话，那么就产生另一种优化分配问题. 设 s_{xy} 表示人 x 对相应诊所中预约时间 y 的评估，这个值与 y 是否在 x 的 10 英里内无关. 这时，我们要寻找人到接种诊所的一种分配，使得这个评估 s_{xy} 的和最大，正如工作分配问题那样(例 12.1). ■

【例 12.8 音响系统中的扬声器配对(Ahuja，Magnanti 和 Orlin[1993]、Mason 和 Philpott[1988])】 音响系统制造商在出售一套音响系统之前要设法把扬声器配对. 如何将两个扬声器配成套取决于频率响应. 两个扬声器 x 和 y 之间的频率响应 f_{xy} 可以(通过比较不同频率的响应)测量，且低 f_{xy} 比高 f_{xy} 要好. 制造商也许只想配对度量 f_{xy} 足够小的扬声器，比如说小于某个值 T. 这时，制造商开始于一个扬声器集合，希望创造尽可能多的扬声器配对. 这是一个*最大基数匹配问题*(maximum-cardinality matching). 扬声器是顶点，x 和 y 之间有边意味着 f_{xy} 小于 T. 更完美的目标是创造满足 $f_{xy}<T$ 的配对，使得这些配对的所有 f_{xy} 的和尽可能小. 这是一个最小权匹配问题. 与例 12.5 的情况对照(在那里最大取代最小)，注意这里的问题是任意图上的问题，没有必要一定是二部图. ■ 686

【例 12.9 石油钻井(Ahuja，Magnanti 和 Orlin[1993]、Devine[1973])】 一家石油公司已勘测出若干可能的钻井场所. 两个场所钻一眼井比分开钻井要廉价. 这家公司估测了在每个场所钻井的成本，以及对每一对场所钻一眼井的成本. 问题是确定哪些场所单独钻井，哪些场所配对钻井，以使总钻井成本最低. 如果我们忽略单独钻井的可能性，那么我们就面对一个最小权匹配问题. 如果我们允许井到其自身有一条边，那么我们还可以阐述整个问题为最小权匹配问题. 注意所讨论的图不是二部图. ■

匹配有很多其他应用. Ahuja，Magnanti 和 Orlin[1993]给出的例子包括确定化学键、在制造业对特定的电子打字机进行重新布线、空间对象定位、匹配移动对象. 最优的存货损耗以及并行机器调度等等. 我们在练习中讨论其中的一些应用.

在 12.7 节中，我们将看到最大权匹配问题与 11.4.1 节的“中国邮差”问题及 11.4.2 节的相关的计算机绘图问题有关. 关于一般的最大权匹配问题以及本章所描述的其他内容，可以参见 Ahuja，Magnanti 和 Orlin[1993]、Christofides[1975]、Cook 等[1998]、Grötschel，Lovász 和 Schrijver[1993]、Lawler[1976]、Lovász 和 Plummer[1986]、Minieka[1978]、Papadimitriou 和 Steiglitz[1982].

本节练习

1. 对于图 12.4 中的图：
   (a)寻找有 4 个边的匹配.
   (b)寻找使顶点 b 饱和的匹配.
   (c)寻找不是最大的但是极大的匹配.
2. 对于图 12.5 中的图，重复练习 1.

687 3. 对于图 12.6 中的每一个加权图，寻找最大权匹配.

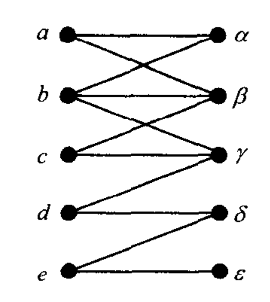


图 12.4 12.1 节练习 1 的图

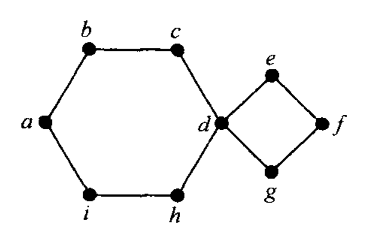
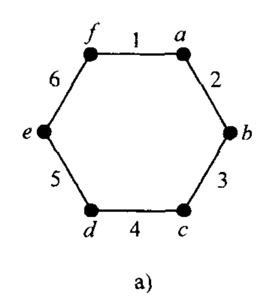
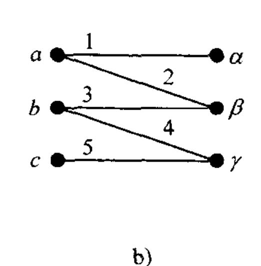


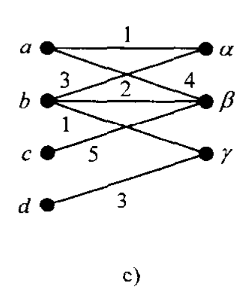
图 12.5 12.1 节练习 2 的图



a)



b)



c)

图 12.6 12.1 节练习 3 的加权图

4. 一家公司有 5 个职位需要填补：计算机技师(c)、接待员(r)、会计师(a)、文字处理操作员(w)和数据分析员(d). 对这些工作有 7 名申请人. 第一名申请人有资格应聘职位 c、r、a；第二名申请人有资格应聘职位 w、d；第三名申请人有资格应聘职位 r、w、d；第四名申请人有资格应聘职位 c、a；第五名申请人有资格应聘职位 c、r、a、w；第六名申请人有资格应聘职位只有 c；第七名申请人有资格应聘所有职位. 所有这些职位空缺都可以填充上吗？使用本节的术语阐述这一问题. 我们要寻找什么？这里不要求你解决这一问题.

5. 一家公司给它的工人分配临时的工作. 一个矩阵的 i，j 项给出工人 i 的家到第 j 项工作的距离. 这个矩阵是所给的信息. 如果每名工人适合每项工作，那么寻找工人到工作的一个分配，这一分配使得工人必须经过的距离总和最小. 用本节的术语这一问题. 不必解决它. 我们要寻找什么？

6. 有六名学生在一个宿舍找房间：Ingram，Knight，Marks，Odell，Quincy，Spencer. Ingram 喜欢 Knight，Marks，Odell，Spencer；Knight 喜欢 Ingram，Odell，Quincy，Spencer；Marks 喜欢 Ingram，Quincy，Spencer；Odell 喜欢 Ingram，Knight，Quincy；Quincy 喜欢 Knight，Marks，Odell；Spencer 喜欢 Ingram，Knight，Marks. 注意，a 喜欢 b 当且仅当 b 喜欢 a. 我们希望两人一间地分配室友，使得每个人都有他喜欢的室友. 我们能给每个人分配一间房间吗？如果不能，我们能够分配房间的最多人数是多少？用本节的术语阐述这一问题. 不必解决它. 我们想要寻找什么？

7.（Ahuja，Magnanti 和 Orlin[1993]和 Brogan[1989]）使用两台红外传感器来确定空间中的物体. 使用每台红外传感器提供一条投射到该物体的光线，这两条光线帮助我们确定该物体的位置. 假设已知从第一台传感器发射 p 条光线，第二台传感器发射 p 条光线，但是不知道哪条光线对应于哪个物体. 从一台传感器发射的光线可以与来自另一台传感器发射的多条光线相交. 同时，由于测量误差，两台传感器对应于相同物体的光线不能相交. 我们希望匹配来自两台传感器的光线. 把这一问题阐述为匹配问题：定义一个图，定义任意适当的测量 r_{xy}，并讨论所涉及的匹配问题的类型：M 饱和，二部图的最大基数匹配等等. 688

8.（Ahuja，Magnanti 和 Orlin[1993]、Brogan[1989]、Kolitz[1991]）在导弹防御和其他应用中，我们希望评估空间物体的速度和方向. 我们取各种色物体在两个不同的时刻的"快照". 如果我们能够匹配对应于同

一物体的第一个快照和第二个快照，那么我们就可以估测它的运动速度(根据两个快照之间的时间)和方向. 设 r_{xy} 表示快照间的距离. 如果我们假设能够非常快地拍摄物体快照，我们可以使用小的 r_{xy} 进行物体匹配. 如练习 7 那样，把这一问题阐述为匹配问题.

9. (Ahuja，Magnanti 和 Orlin[1993]、Derman 和 Klein[1959])在我们存货(书、化学制品等)目录中的产品，经过一段时间后或者升值或者减值. 假设在开始时，我们知道存货目录中每一种产品的年代. 假设我们有一个时间集合，在集合中的每个时间，我们必须从我们的存货目录中提取一个产品用于某种过程，诸如制作过程. 我们应该什么时候提取给定产品呢？假设我们可以使用效用函数 $u(a)$ 来评估年代 a 的产品的价值. 那么，在这个过程的一开始，我们可以评估在时间 t_j 提取年代 a_i 的产品的效用 u_{ij}. 如练习 7 那样，把求解从存货目录中提取产品的计划问题阐述为匹配问题.
10. 证明：为了描述加权图 G 的最大权匹配问题的一般解，我们可以假设：
   (a) G 是有偶数个顶点.
   (b) G 是完全图.
11. 证明：为了描述二部图 G 的最大权匹配问题的一般解，我们可以假设这个图是完全二部图且两个类有相同数量的顶点.
12. 证明：一个加权二部图的最大权匹配不必使第一类顶点饱和，即使该类中的顶点不超过一半.
13. 画出一个 $|X|=|Y|=3$ 的二部图 $G=(X, Y, E)$，使得 G 有唯一 X 饱和匹配且 $|E|$ 尽可能大. 解释为什么不存在更大的 G.
14. (Ahuja，Magnanti 和 Orlin[1993])滑雪租赁店希望把 n 对滑雪板分配给 n 位滑雪者. 滑雪板有长度 $l_1 \leqslant l_2 \leqslant \cdots \leqslant l_n$，滑雪者的身高为 $h_1 \leqslant h_2 \leqslant \cdots \leqslant h_n$. 理想地，身高 h 的滑雪者得到长度为 αh 的滑雪板，其中 α 是固定的常量. 然而，给定滑雪板的供给是有限的，这家店只能设法最小化 αh_i 与分配给滑雪者 i 的滑雪板的长度之间的(绝对)差的和. 证明：如果按上述的那样标记滑雪板和滑雪者，那么把第 i 对滑雪板分配给第 i 个滑雪者是最优的.

689

12.2　一些存在结果：二部匹配和相异表示系统

12.2.1　二部匹配

在前一节的很多例子中，我们构造的是二部图 $G=(X, Y, E)$. 这里，我们要问这样的问题：给定二部图 $G=(X, Y, E)$，在什么样的条件下存在使 X 饱和的匹配？

考虑图 12.7 的计算机程序存储分配图. 注意，对于三个程序 a，c，e，一共只有两个可能的存储位置，位置 1 和位置 3. 因此，不存在把每一个程序指定给一个有充分存储容量的位置的存储分配. 不存在使 X 饱和的匹配. 扩展这个例子，显然，为使存在 X 饱和匹配，当 S 是 X 的任意顶点集合而 $N(S)$ 是 S 的*开邻域*(open neighborhood)，即 $N(S)$ 是通过一条边与 S 中的某个 x 连接的所有顶点 y 的集合时，那么 $N(S)$ 的元素数目至少与 S 的元素数目相同. 在我们的例子中，S 是 $\{a, c, e\}$，而 $N(S)$ 是 $\{1, 3\}$. 令人吃惊的是，这个显然的必要条件也是充分的.

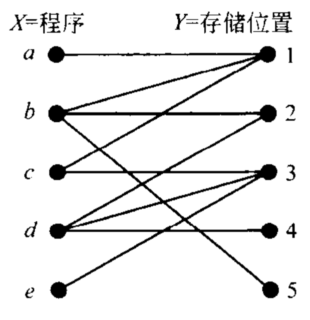


图 12.7　计算机程序存储问题的二部图. X 中的顶点是程序，Y 中的顶点是存储位置，X 中的 x 与 Y 中 y 之间存在一条边当且仅当 y 有足够存储 x 的容量

定理 12.1 (Philip Hall 定理，Hall[1935])　*设 $G=(X, Y, E)$ 是二部图. 那么存在使 X 饱和的匹配当且仅当，对于 X 的所有子集 S，有 $|N(S)| \geqslant |S|$.*

我们将在 12.4 节证明定理 12.1.

为了看清图 12.1 中的二部图是否有使 X 饱和的匹配，其中 $X=\{\text{Smith, Jones, Brown, Black, White}\}$，我们必须对 X 的所有子集 S 计算 $N(S)$. 有 $2^{|S|}=2^5=32$

个这样的子集．为了计算，例如，我们注意到

690

$$N(\{\text{Smith, Jones, Black}\})=\{切割，定型，粘合，包装\},$$

所以 $N(S)$ 有四个元素，而 S 有三个元素．对于所有 32 种情况的类似计算表明存在使 X 饱和的匹配（在这种情况下，直接寻找一个使 X 饱和的匹配更快）．

【例 12.10 编码问题】 一个小文件中的每一项都有一个与它相关的 3 数字代码字，使用的数字是 0，1，2，…，n．我们能否把这个文件中的每一项正好与这一项的代码字中的一个数字联系起来，使得可以根据这一单一数字唯一地恢复这个项？我们可以如下从图论的角度阐述这一问题．一个二部图 $G=(X, Y, E)$ 的顶点是由项的集合 X 和数字 0，1，2，…，n 的集合 Y 组成的．当在项 x 的代码字中使用数字 y 时，有一条从 x 到 y 的边．图 12.8 给出一个例子．我们在二部图 G 中寻找一个使 X 饱和的匹配．在我们的例子中，表 12.1 列出 X 的所有子集，并列出对应的开邻域 $N(S)$．注意，对于每一种情况都有 $|N(S)|\geqslant|S|$．因此，存在使 X 饱和的匹配．这并不能帮助我们寻找使 X 饱和的匹配．然而，容易看到边 $\{a, 3\}$，$\{b, 0\}$，$\{c, 1\}$，$\{d, 2\}$ 形成一个这样的匹配．

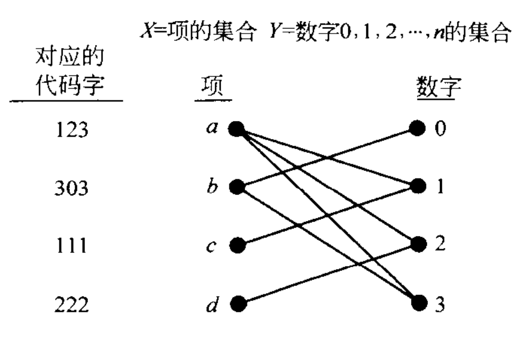


图 12.8 $n=3$ 时的编码问题的二部图

表 12.1 对应于图 12.8 的 X 的子集 S 及对应的开邻域 $N(S)$

S	$\varnothing$	$\{a\}$	$\{b\}$	$\{c\}$	$\{d\}$	$\{a, b\}$
$N(S)$	$\varnothing$	$\{1, 2, 3\}$	$\{0, 3\}$	$\{1\}$	$\{2\}$	$\{0, 1, 2, 3\}$
S	$\{a, c\}$	$\{a, d\}$	$\{b, c\}$	$\{b, d\}$	$\{c, d\}$	$\{a, b, c\}$
$N(S)$	$\{1, 2, 3\}$	$\{1, 2, 3\}$	$\{0, 1, 3\}$	$\{0, 2, 3\}$	$\{1, 2\}$	$\{0, 1, 2, 3\}$
S	$\{a, b, d\}$	$\{a, c, d\}$	$\{b, c, d\}$	$\{a, b, c, d\}$		
$N(S)$	$\{0, 1, 2, 3\}$	$\{1, 2, 3\}$	$\{0, 1, 2, 3\}$	$\{0, 1, 2, 3\}$		

■

一个图 G 称为是正则的（regular），如果它的每个顶点都有相同的度，即每个顶点的邻居数量相同．

691

推论 12.1.1 *假设 $G=(X, Y, E)$ 是至少有一条边的正则二部图．那么 G 有使 X 饱和的匹配．*

证明 设 S 是 X 的子集．设 E_1 是从 S 的顶点出发的边的集合，而 E_2 是从 $N(S)$ 的顶点出发的边的集合．因为 E_1 的每一条边都从 S 的顶点指向 $N(S)$ 的顶点，所以 $E_1\subseteq E_2$．因此，$|E_1|\leqslant|E_2|$．另外，因为每一个顶点都有相同数量的邻居，比如说有 k 个邻居，又因为 $S\subseteq X$，所以有 $|E_1|=k|S|$ 和 $|E_2|=k|N(S)|$ 成立．因此 $k|S|\leqslant k|N(S)|$．因为 G 有一条边，k 是正数，所以我们得出结论 $|S|\leqslant|N(S)|$． ■

作为推论 12.1.1 的一个应用，假设在例 12.10 中，每一个代码字正好使用数 0，1，2，…，n 中的 k 个数字，而且 0，1，2，…，n 中的每一个数字正好在 k 个代码字中出现．那么文件中的每个项都有使我们有唯一解码的数字．

注意，正则二部图 $G=(X, Y, E)$ 的使 X 饱和的匹配也使 Y 饱和，因为 G 的每一条边既邻接 X 的顶点又邻接 Y 的顶点．因此，如果 k 是每个顶点的邻居数量，那么 $k|X|=|E|$ 和 $k|Y|=|E|$ 成立．因此，$k|X|=k|Y|$ 和 $|X|=|Y|$ 成立．因此，每个使 X 饱和的匹配一定使 Y 饱和．使一个图的每个顶点都饱和的匹配称为完美的（perfect）．工作分配中的完美匹配意味着，每名

工人得到一项工作，而每项工作有一名工人来做. 12.3 节进一步讨论，在一般情况下，什么时候存在完美匹配的问题(也可参见练习 18、练习 21、练习 24 和练习 26).

12.2.2 相异表示系统

假设$\mathcal{F}=\{S_1, S_2, \cdots, S_P\}$是一个集合族，各集合不必不同. 设 $T=(a_1, a_2, \cdots, a_p)$是一个 p 元组，其中 $a_1 \in S_1$，$a_2 \in S_2$，…，$a_p \in S_p$. 这时，T 称为$\mathcal{F}$的表示系统(system of representative). 另外，如果所有的 a_i 各不相同，那么 T 称为$\mathcal{F}$的相异表示系统(system of distinct representative)(SDR). 例如，假设

$$\mathcal{F} = \{S_1, S_2, S_3, S_4, S_5\},$$

其中

$$S_1 = \{a,b,c\},\quad S_2 = \{b,c,d\},\quad S_3 = \{c,d,e\},\quad S_4 = \{d,e\},\quad S_5 = \{e,a,b\}. \tag{12.1}$$

那么 $T=(a, b, c, d, e)$是$\mathcal{F}$的 SDR. 接下来，如果$\mathcal{F}=\{S_1, S_2, S_3, S_4, S_5, S_6\}$，其中各 S_i如下所示：

$$\begin{aligned} &S_1 = \{a,b\},\quad S_2 = \{b,c\},\quad S_3 = \{a,b,c\}, \\ &S_4 = \{b,c,d\}, S_5 = \{a,c\},\quad S_6 = \{c,d\}. \end{aligned} \tag{12.2}$$

那么 $T=(a, b, c, d, a, d)$是$\mathcal{F}$的一个表示系统. 然而，正如我们下面所看到的那样，不存在 SDR.

692

【例 12.11 列表着色(再探例 3.22)】 在例 3.22 中，我们介绍了图 G 的列表着色的概念，即 G 对于每个顶点 x 有一个可接受颜色列表 $L(x)$的列表分配. 一个列表着色是一个普通的图着色，其中指定给顶点 x 的颜色是从 $L(x)$中选择的. 如果 G 是完全图，那么列表着色对应于列表 $L(x)$的 SDR. ■

【例 12.12 医院实习医生分配】 在给定年，假设有 p 名医学院毕业生申请做医院实习医生. 对于第 i 名医学院毕业生，设 S_i 是可接收 i 的所有医院的集合. 这时，集合族$\mathcal{F}=\{S_i: i=1, 2, \cdots, p\}$的表示系统为每名实习医生指定一家愿意接收他的医院. 另外，一个 SDR 将保证没有医院会得到超过一名的实习医生. 实际上可用下面的方法使用 SDR. 修改 S_i，使其包含 i. 寻找一个 SDR. 这个 SDR 为每个 i 指定一家医院或他自己；后者表示在第一轮没有给 i 指定医院(练习 7 要求读者证明 SDR 的存在性). 在 SDR 的最初分配之后，要求各家医院修改还没有分配出去的可接收申请者的列表(理想的是，在第一轮至少有一些人被分配出去). 然后，寻找一个新的 SDR；以此类推. 安排申请者去研究院的入学许可也使用类似的过程. 从 1952 年开始，在全国居民匹配程序中所使用的更复杂的过程就利用了医院及申请人分级，或选择分级. 我们将在 12.7 节讨论这一思想. ■

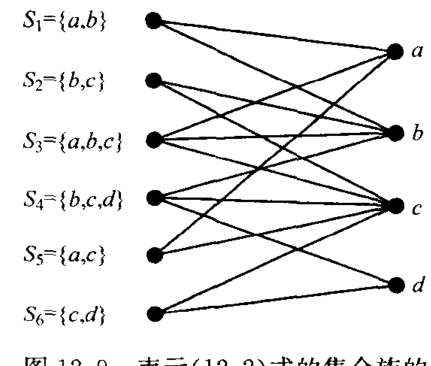


图 12.9 表示(12.2)式的集合族的二部图

寻找 SDR 的问题可以从图论的角度阐述. 假设 X 是$\mathcal{F}$中的集合 S_i的集合，Y 是 $\bigcup S_i$ 中的点的集合. 设 $G=(X, Y, E)$是一个二部图，对于 Y 中的 y 和 X 中的 x，当且仅当 y 在 x 中时，从 x 到 y 存在一条边. 这时，SDR 就是 G 中的使 X 饱和的匹配.

例如，考虑(12.2)式中的集合族. 图 12.9 给出对应的二部图. 注意，如果 $S=\{S_1, S_2, S_3, S_5\}$，那么 $N(S)=\{a, b, c\}$. 因此 $|S|>N(S)$，所以根据 Philip Hall 定理，不存在使 X 饱和的匹配，因此，不存在 SDR. 使用 SDR 的术语，现在我们可以如下重新阐述 Philip Hall 定理.

推论 12.1.2　集合族$\mathcal{F}=\{S_1, S_2, \cdots, S_p\}$拥有 SDR，当且仅当对于所有的$k=1, 2, \cdots, n$，任意$k$个$S_i$总共包含至少$\bigcup S_i$的$k$个元素.

【例 12.13 扩展拉丁矩形】　拉丁矩形(latin rectangle)是使用元素 1，2，…，n的$r\times s$数组，使得每一行每一列使用 1，2，…，n的元素至多一次. 一个拉丁矩形称为是完全的(complete)，如果$n=s$. 构建$n\times n$拉丁方的一种方法是尝试着一次构建一行，即首先创建$r\times n$的完全拉丁矩阵，然后再扩展它. 例如，考虑图 12.10 中所示的 2×6 拉丁矩形. 可以把它扩展成 6×6 拉丁方吗？更特殊地，我们可以在这个拉丁矩形上增加第三行而得到 3×6 的拉丁矩形吗？解决这一问题的一种方法是使用车多项式(参见 5.1 节练习 15). 另一种方法是问：在新行上的第i列可以放置什么数字. 设S_i是还没有出现在第i列上的数字的集合. 在我们的例子中，

$$S_1=\{2,3,5,6\},\quad S_2=\{1,4,5,6\},\quad S_3=\{1,2,4,5\},$$
$$S_4=\{1,2,3,6\},\quad S_5=\{2,3,4,6\},\quad S_6=\{1,3,4,5\}.$$

我们要从每个S_i中选出一个元素，并要求这些元素不同. 因此，我们要求一个 SDR. 一个 SDR 是(2，1，4，3，6，5). 因此，我们可以使用这一组数字作为新的第三行，得到如图 12.11 所示的拉丁矩形. 可以把这一想法一般化. 一般地，给定$r\times n$完全拉丁矩形，设S_i是还没有在第i列出现的数字的集合. 为了增加第$(r+1)$行，我们需要一个S_i族的 SDR. 我们将证明总可以找到一个 SDR.

1	2	3	4	5	6
4	3	6	5	1	2

图 12.10　2×6 拉丁矩形

1	2	3	4	5	6
4	3	6	5	1	2
2	1	4	3	6	5

图 12.11　通过扩展图 12.10 中的矩形而得的 3×6 拉丁矩形

■

定理 12.2　如果$r<n$，那么每个$r\times n$完全拉丁矩形L可以扩展成一个$(r+1)\times n$完全拉丁矩形.

证明[⊖]　如果S_i如例 12.13 中所定义的那样，那么我们要证明集合族S_i拥有 SDR. 在这个集合族中选出k个集合$S_{i1}, S_{i2}, \cdots, S_{ik}$. 根据推论 12.1.2，只要证明$A=S_{i1}\cup S_{i2}\cup\cdots\cup S_{ik}$至少有$k$个元素即可. 每个$S_{ij}$有$n-r$个元素. 因此，$A$有$k(n-r)$个元素，其中包含重复的元素. 因为$r\times n$拉丁矩形$L$是完全的，所以每个来自于 1，2，…，$n$的数字正好在每一行上出现一次，所以 1，2，…，$n$中的每个数字在$L$中正好出现$r$次. 因此，来自 1，2，…，$n$中的每个数字正好在$n-r$个集合$S_1, S_2, \cdots, S_n$中. 因此，$A$中的每个数字至多在$n-r$个集合$S_{i1}, S_{i2}, \cdots, S_{ik}$中出现. 现在，如果我们列出$A$的所有元素，包括重复的，那么我们有$k(n-r)$个元素. A中的每个数字在这个列表中至多出现$n-r$次. 根据鸽巢原理(定理 2.15)，在A中至少存在k个不同的数字. ■

本节练习

1. 对于下面的图G和集合S，寻找$N(S)$.
   (a)G是图 12.1 所示的图，$S=\{\text{Smith, White}\}$.
   (b)G是图 12.8 所示的图，$S=\{a, c, d\}$.
   (c)G是图 12.9 所示的图，$S=\{S_1, S_3, S_4\}$.
2. 对于下面的图G和集合S，寻找$N(S)$.
   (a)$G=$图 12.1 所示的图，$S=\{$切割，定型$\}$.

⊖　本证明可以跳过.

(b)G=图 12.1 所示的图，S={Smith，Black，Polishing}.

(c)G=图 12.5 所示图，$S=\{a, b, e\}$.

3. 对于图 12.12 中的每一个二部图 $G=(X, Y, E)$，确定 G 是否有使 X 饱和的匹配.

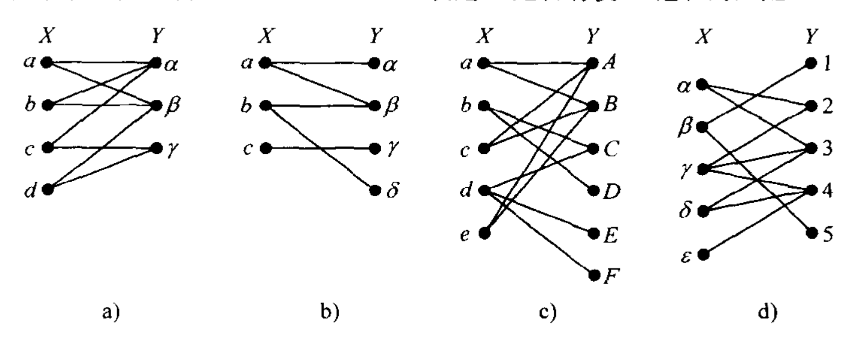


图 12.12　12.2 节练习的二部图

4. (a)寻找对应于(12.1)式的集合族的二部图.

   (b)使用 Philip Hall 定理证明(a)的二部图中存在使 X 饱和的匹配.

5. 对于下面每一个集合族，寻找一个表示系统.

   (a)$S_1=\{a, b, f\}$，$S_2=\{a\}$，$S_3=\{a, b, d, f\}$，$S_4=\{a, b\}$，$S_5=\{b, f\}$，$S_6=\{d, e, f\}$，$S_7=\{a, f\}$

   695

   (b)$S_1=\{a, c\}$，$S_2=\{a, c\}$，$S_3=\{a, c\}$，$S_4=\{a, b, c, d, e\}$

   (c)$S_1=\{a_1, a_2, a_3\}$，$S_2=\{a_3, a_4\}$，$S_3=\{a_1, a_2, a_3\}$，$S_4=\{a_2, a_4\}$

   (d)$S_1=\{b_3, b_5\}$，$S_2=\{b_1, b_3, b_4\}$，$S_3=\{b_3, b_4, b_5\}$，$S_4=\{b_3, b_4, b_5\}$

   (e)$S_1=\{x, y\}$，$S_2=\{x\}$，$S_3=\{u, v, w\}$，$S_4=\{x, y, z\}$，$S_5=\{y, z\}$

   (f)$S_1=\{a, b, c\}$，$S_2=\{b, c\}$，$S_3=\{c, e, f\}$，$S_4=\{a, b\}$，$S_5=\{a, c\}$，$S_6=\{d, e, f\}$

6. 对于练习 5 中的每一个集合族，确定是否存在相异表示系统，如果存在，寻找一个这样的系统.

7. 在例 12.12 中，证明如果 i 在 S_i 内，那么存在 SDR.

8. 求下面每一个集合族的 SDR 的数量.

   (a)$S_1=\{1, 2\}$，$S_2=\{2, 3\}$，$S_3=\{3, 4\}$，$S_4=\{4, 5\}$，$S_5=\{5, 1\}$，$S_6=\{6, 1\}$

   (b)$S_1=\{1, 2\}$，$S_2=\{2, 3\}$，$S_3=\{3, 4\}$，…，$S_n=\{n, 1\}$

   (c)$S_1=\{1, 2\}$，$S_2=\{3, 4\}$，$S_3=\{5, 6\}$，$S_4=\{7, 8\}$，$S_5=\{9, 10\}$

   (d)$S_1=\{1, 2\}$，$S_2=\{3, 4\}$，$S_3=\{5, 6\}$，…，$S_n=\{2n-1, 2n\}$

9. 有六个国家立法委员会：财政、环境、健康、运输、教育和住房. 在第 1 章，我们讨论了这些委员会的会议安排问题. 这里，我们要问另外一个不同的问题. 假设有 12 个立法者，需要把他们安排到委员会中，每个人去一个委员会. 下面的矩阵的 i，j 项等于 1 当且仅当第 i 个立法者喜欢就职于第 j 个委员会.

	财政	环境	健康	运输	教育	住房
Allen	1	1	1	0	0	0
Barnes	1	1	0	1	1	0
Cash	1	1	1	0	0	0
Dunn	1	0	0	1	1	1
Ecker	0	1	1	0	0	0
Frank	1	1	0	0	0	0
Graham	1	1	1	0	0	0
Hall	1	0	0	0	0	0
Inman	1	1	1	0	0	0
Johnson	1	1	0	0	0	0

假设我们要为每一个委员会正好选出一个新成员，只选择喜欢该职业的立法者. 你能做到这一点吗?

（不是每一个立法者都必须被指定一个委员会，且不能把一个立法者指定给多个委员会.）

10. 考虑顶点$\{a, b, c, d, e, f\}$的完全图，且有列表$L(a)=\{2, 3, 4\}$，$L(b)=\{3, 4\}$，$L(c)=\{1, 4, 6\}$，$L(d)=\{2, 3\}$，$L(e)=\{2, 4\}$，$L(f)=\{1, 5, 6\}$. 存在列表着色吗？

11. 假设$G=\{X, Y, E\}$是一个完全二分图，且对于$X\cup Y$中的每一个顶点a有列表$L(a)$. 给出证明或者举出反例：G有L列表着色当且仅当列表$L(a)$的集合有SDR.

12. 图G的一个**边着色**(edge coloring)是给G的每一条边指定一种颜色，使得如果两条边有公共顶点，那么这两条边得到不同的颜色.

696

(a)证明：有k种颜色的边着色把图分解成k个边独立匹配.

(b)证明：如果二部图G是正则的，且每个顶点的度为k，那么存在有k种颜色的边着色.

(c)证明：如果G是一个二部图且$\Delta(G)$是G中顶点的最大度数，那么G有使用$\Delta(G)$种颜色的边着色［提示：通过加入顶点和边使得G成为正则图］.

13. 如果q是一个正整数且$S_1, S_2, \cdots, S_p$是整数的集合，我们说一个SDR$(a_1, a_2, \cdots, a_p)$是**q距离表示系统**(q-distant representative)，如果对于所有的$i\neq j$有$|a_i-a_j|\geqslant q$. 在信道分配中出现的这一概念是由Fiala，Kratochvil和Proskurowski[2001]给出的. 假设G是完全图，对于G的每一个顶点a有整数列表$L(a)$. 证明：对于某一个集合T，G有T着色(练习3.20)，当且仅当列表$L(a)$的集合有q距离表示系统.

14. 假设我们有一个代码字集合，而且希望把这个代码字集合尽可能简单地存储在计算机中，即从每一个代码字中选出一个字母存储起来.

(a)使用代码字被唯一解码的方法，对下面的代码字集合能完成这一工作吗？字：*abcd*，*cde*，*a*，*b*，*ce*.

(b)如果能够完成，我们可以用多少种方法做这件事？

15. 求解12.1节练习4.

16. 考虑下面的数组. 对于每一种情况，能否使用数字1，2，3，4，5，6，7给这个数组再增加一列，使得在新的一列中的各项都是不同的，而且在新数组的各行中所有项也都是不同的？



17. 给出证明或举出反例：至少有一条边的每一个正则二部图有完美匹配.

18. (a)哪些回路Z_n有完美匹配？

(b)哪些有n个顶点的链L_n有完美匹配？

(c)通过给Z_n的所有顶点增加一个邻接顶点可以生成一个轮图W_n. 哪些轮图W_n有完美匹配？

(d)哪些树有完美匹配？

19. 有n名男士和n名女士，且每一名男人正好喜欢p名女人，而每一名女士正好喜欢p名男士. 假设a喜欢b当且仅当b喜欢a，证明$2n$名人可配对成n对夫妇，使得相互喜欢的男士与女士配成夫妇.

20. 如果每一名男士至少喜欢p名女士，而每一名女士至多喜欢p名男士，练习19的结论还成立吗？为什么？

697

21. (Tucker[1980])有n名男士和n名女士在计算机约会服务中心登记. 假设这一服务中心已经做了nm个配对，使得每一男士与m名不同的女士约会，而每一名女士与m名不同的男士约会，假设$m<n$.

(a)假设我们想要在m个晚间安排所有这些约会. 证明有可能做到这一点. 即证明这些配对可以被分成m个完美匹配.

(b)证明我们可以每次一步进行(a)的分配. 即证明无论前k个完美匹配是如何选择的，我们总是可以寻找到第$(k+1)$个完美匹配.

22. 设$\mathcal{F}=\{S_1, S_2, \cdots, S_n\}$是有SDR的集合族.
(a)如果x至少在一个S_i中，证明存在SDR，它选择x为至少一个S_i的表示.
(b)对于每一个包含x的S_i，一定存在选择x作为S_i的表示的SDR吗？为什么？
23. 设$\mathcal{F}=\{S_1, S_2, \cdots, S_n\}$，其中$S_i=\{1, 2, \cdots, n\}-\{i\}$
(a)证明$\mathcal{F}$有SDR.
(b)证明不同SDR的数量是n个元素的错位排列的数量D_n.
24. 一棵树有可能有多个完美匹配吗？为什么？
25. 考虑二部图$G=(X, Y, E)$. 假设$m\geqslant 1$是G的顶点的最大度数，且X_1是由所有度数为m的顶点组成的X的子集. 假设X_1是非空的. 给出下面陈述的证明或给出反例：存在G的匹配，在这个匹配中X_1的所有顶点是饱和的.
26. (a)如果H是一个图，设$o(H)$计数有奇数个顶点的H的分支数量. 如果S是图G的顶点的一个集合，G-S是由不在S中的顶点生成的G的子图. 证明：如果G有完美匹配，那么对于所有的$S\subseteq V(G)$，有$o(G\text{-}S)\leqslant |S|$(Tutte[1947]证明了此结果及其逆叙).
(b)(Peterson[1891])扩展11.2节的想法. 令图G中的一条边$\{a, b\}$称为一个桥(bridge)，如果删除这条边，但不删除它端部的顶点a和b，将增加分支的数量. 假设G是一个图，在这个图中每一个顶点有度数3，并假设G没有任何桥. 假设(a)的逆叙，证明G有完美匹配.
27. 假设$A_1, A_2, \cdots, A_m$和$B_1, B_2, \cdots, B_m$是同一集合S的两种划分，A_i和B_j的数量相同，所有$A_i\neq\varnothing$，所有$B_j\neq\varnothing$(S的划分就是把S的所有元素分成不相交的子集). 设E是S的m元素子集，使得对于所有的i有$A_i\cap E\neq\varnothing$，对于所有的j有$B_j\cap E\neq\varnothing$. 那么，显然，对于所有的i，j，有$|A_i\cap E|=|B_j\cap E|=1$. 集合$E$称为划分$A_i$和$B_j$的*公共表示系统*(system of common representative)(SCR).
(a)对于下列$S=\{1, 2, 3, 4, 5, 6\}$的划分寻找SCR：$A_1=\{1, 2, 3\}$，$A_2=\{4\}$，$A_3=\{5, 6\}$；$B_1=\{1, 3\}$，$B_2=\{2, 6\}$，$B_3=\{4, 5\}$.
(b)证明：划分A_i和B_j有SCR，当且仅当存在划分A_i和B_j的适当重编号，使得对于所有i有$A_i\cap B_i\neq\varnothing$.
(c)证明：划分A_i和B_j有SCR，当且仅当对于所有的$k=1, 2, \cdots, m$及所有的$i_1, i_2, \cdots, i_k, j_1, j_2, \cdots, j_{k-1}$，集合$A_{i_1}\cap A_{i_2}\cup\cdots\cap A_{i_k}$不是集合$B_{j_1}\cup B_{j_2}\cup\cdots\cup B_{j_{k-1}}$的子集. [提示：构建一个二部图并构建一个匹配.]

698

12.3 任意图的完美匹配的存在性[⊖]

【例12.14 英国皇家空军飞机的飞行员(再探例12.4)】 让我们考虑给定一组飞行员，是否可能给每一架飞机指定两名相容的飞行员，使得每一名飞行员被指定一架飞机. 情况是这样的，当且仅当例12.4所构造的图有完美匹配，在这一匹配中每一个顶点都是饱和的. 本节研究保证完美匹配存在的一个条件. ■

定理12.3 *如果图G有$2n$个顶点且每个顶点的度数$\geqslant n$，那么这个图有完美匹配.*

我们通过给出构建匹配M的步骤来证明为什么这一定理为真. 下面是构建匹配M的算法.

算法12.1：寻找完美匹配

输入： 有$2n$个顶点且每个顶点的度数$\geqslant n$的图G.

输出： G的一个完美匹配.

步骤1. 设置$M=\varnothing$.

步骤2. 寻找任意一对由G的边连接的不饱和顶点a和b，把边$\{a, b\}$放入M中.

⊖ 本小节可以跳过而不失连续性.

步骤 3. 如果存在由 G 的边连接的一对 G 的不饱和顶点，返回步骤 2. 否则，前进到步骤 4.

步骤 4. 如果 M 有 n 条边，停止并输出 M. 如果 M 没有 n 条边，前进到步骤 5.

步骤 5. 寻找 G 中的一对不饱和顶点 a 和 b. 它们不被 G 的边连接(为什么). 寻找 M 的边 $\{u, v\}$，使得 $\{a, u\}$ 和 $\{b, v\}$ 都是 G 的边. 从 M 中删除边 $\{u, v\}$ 并把边 $\{a, u\}$ 和 $\{b, v\}$ 放入 M 中. 返回步骤 4.

699

我们使用图 12.13 的图说明这一过程. 我们将在后面证明为什么这一过程可行. 注意，这个图中有 $2n=2(3)=6$ 个顶点，而且每个顶点的度数 $n\geqslant 3$. 在步骤 2 中，从把边 $\{1, 6\}$ 放入 M 中开始. 表 12.2 给出 M 的各个边. 前进到步骤 3. 接下来，因为存在一对由 G 的边连接的不饱和顶点，返回到步骤 2. 寻找这样的一对顶点，比如说是 3 和 5，并把边 $\{3, 5\}$ 放入到 M 中. 现在，不存在任何一对由 G 的边连接的不饱和的顶点，而且因为 M 的边小于 n，所以我们从步骤 3 前进到步骤 4，再到步骤 5. 选出两个顶点 2，4，它们在 G 中不饱和，且没有被 G 的边连接. 注意，在 M 中 3 和 5 是匹配的，而且存在 G 中的边 $\{2, 3\}$ 和 $\{4, 5\}$. 因此从 M 中删除边 $\{3, 5\}$，加入边 $\{2, 3\}$ 和 $\{4, 5\}$，返回到步骤 4. 因为在 M 中我们有 $n=3$ 个元素，我们停止，得到一个完美匹配.

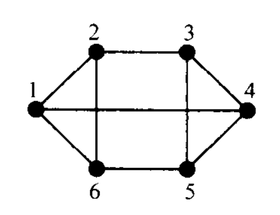


图 12.13 用于说明算法 12.1 的图

表 12.2 将算法 12.1 运用于图 12.13 中的图时每一步中 M 的边

步骤	1	2	3	4
M 中的边	无	{1, 6}	{1, 6}, {3, 5}	{1, 6}, {2, 3}{4, 5}

为了弄明白为什么这一过程可行，注意，在每一个阶段，我们都有一个匹配. 另外，如果这一算法停止，那么我们总以 M 中的 n 条边结束，因为在 G 中存在 $2n$ 个顶点，所以我们必定有一个完美匹配. 关键的问题是：我们如何知道步骤 5 可行？特别地，我们如何知道总存在步骤 5 需要的两个顶点 u 和 v？为了明白其原因[⊖]，假设 a 和 b 在 M 中是不饱和的，而且不被 G 的边连接. 此时在 M 中存在 $r<n$ 条边. 如果步骤 5 不能被运行，那么对于 M 中的每一条边 $\{u, v\}$，边 $\{a, u\}$，$\{a, v\}$，$\{b, u\}$，$\{b, v\}$ 中至多有两条边在 G 中. 在 G 中从 a 或 b 到匹配的边 $\{u, v\}$ 的边的数量因此至多是 $2r$. 从 a 或从 b 的每一条边都经过 M 中的某条边，因为否则 a 或 b 就会通过 G 的一条边与 G 的一个不饱和顶点连接，那么我们就无法进行到此算法的步骤 5. 因此，a 的度数 $+b$ 的度数 $\leqslant 2r$. 因为 $r<n$，所以 a 的度数 $+b$ 的度数 $<2n$. 但是根据假设，a 的度数 $+b$ 的度数 $\geqslant n+n=2n$，这是一个矛盾. 于是我们得出结论，此算法总是可以运行步骤 5.

700

我们来检查一下算法 12.1 的计算复杂度. 为了得到一个大致的上界，假设这一算法的第 i 次迭代是匹配 M 从有 $i-1$ 条边到有 i 条边之间的过程. 因此，总共有 n 次迭代. 我们可以记录不饱和顶点. 因此，寻找连接一对不饱和顶点的边至多花费 e 步，其中 $e=|E(G)|$. 如果没有这样的边，我们选出任意一对不饱和的顶点 a 和 b. 对于 M 中的每一条边 $\{u, v\}$，我们问是否 $\{a, u\}$ 和 $\{b, v\}$ 都是 G 的边，或 $\{a, v\}$ 和 $\{b, u\}$ 是否是 G 的边. 四个问题中每一个问题至多需要 e 条边，所以至多需要 $4e$ 条边. 因此，每一次迭代至多有 $e+4e=5e$ 步，而且总共有 $n(5e)$ 步. 因为 $e\leqslant\binom{n}{2}\leqslant n^2$，所以我们至多有 $5n^3$ 步，这是一个多项式界. 使用 2.18 节的术语，我们有一个 $O(n^3)$ 的算法.

[⊖] 本讨论可以跳过.

本节练习

1. 假设 K_{2p}是有 $2p$ 个顶点的完全图，$p\geqslant 2$，且 G 是通过从 K_{2p}中删除一条边而得到的. G 有完美匹配吗？
2. 完全 p 部图(complete p-partite graph)$K(n_1, n_2, \cdots, n_p)$有 p 组顶点，在第 i 类中有 n_i 个顶点，对于 $i\neq j$，第 i 类中的所有顶点与第 j 类中的所有顶点相连接，且不与其他顶点连接. 下面各图中哪些有完美匹配？

   (a)$K(2, 2)$　(b)$K(3, 3)$　(c)$K(1, 5)$　(d)$K(2, 3)$

   (e)$K(2, 2, 2)$　(f)$K(3, 3, 3)$　(g)$K(4, 4, 4)$　(h)$K(2, 2, 2, 2)$
3. 在 2.1 节练习 6 中，能给所有学生指定室友吗？
4. 对于图 12.14 中的每一个图，使用算法 12.1 寻找一个完美匹配.

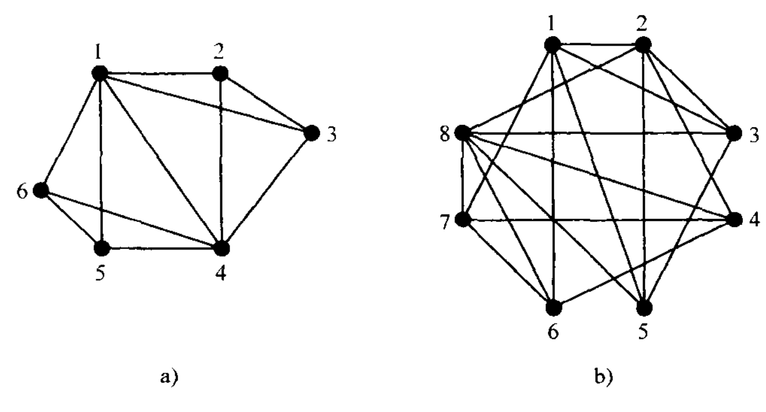


图 12.14　12.3 节和 12.4 节练习的图

5. 假设 G 有 $2n$ 个顶点，对于所有使得$\{a, b\}$不是 G 的边的 $a\neq b$，a 的度数$+b$ 的度数$\geqslant 2n$. 证明 G 有完美匹配.

701

12.4 最大匹配和最小覆盖

12.4.1 顶点覆盖

一个图中的*顶点覆盖*(vertex covering)，或简称*覆盖*(covering)，就是包含这个图的每一条边的至少一个端点的顶点集合. 例如，对于图 12.13 中的图，顶点集合$\{1, 2, 3, 4, 5\}$形成一个覆盖. 而顶点集合$\{1, 3, 5, 6\}$也形成一个覆盖. 本节研究*最小基数覆盖*(minimum cardinality covering，简称*最小覆盖*，minimum covering). 这是尽可能使用少量顶点的覆盖. 我们将讨论最小覆盖与*最大基数匹配*(maximum cardinality matching，简称*最大匹配*，maximum matching)之间的关系. 例 12.4 中出现过这样的匹配.

【例 12.15 警察监视】 站在城市交叉路口的警官能够观察到任意方向的一个街区. 如果我们希望定位一名罪犯的位置，那么我们要肯定附近的所有街区都在观察的范围之内. 假设警官们只站在交叉路口，我们所需要的警官的最小数量是多少，我们要把他们安排在什么地方？通过设附近的交叉路口是一个图 G 的顶点，并且用一条边连接两个顶点，当且仅当两个交叉路口由一个街区街道连接，我们就可以回答上述问题. 这时，我们要寻找这个图 G 的最小覆盖. ■

【例 12.16 天花疫苗接种(再探例 12.3)】 在例 12.3 中，我们研究了把人们分配到距离他们的家不超过 10 英里的接种诊所的分配. 在那里，我们假设接种诊所已经确定. 但是，如果我们想要定位尽可能少的诊所，使得每一个人都在某一诊所的 10 英里范围内. 我们可以设这个城市的每一个方形街区是一个顶点，如果两个方形街区彼此在 10 英里内，则用一条边连接这两个顶

点．假设每一个方形街区都有人居住，那么我们寻找一个数目最少的顶点（方形街区）集合，使得每一条边都有端点在这个集合内．这样的集合是这个图的一个最小覆盖．■

接下来，我们要研究匹配与覆盖之间的关系．设 G 是一个图，M 是 G 的一个匹配，设 K 是一个覆盖．那么 $|M| \leqslant |K|$，因为已知 M 的任意边 $\alpha=\{x, y\}$，或者 x 在 K 中，或者 y 在 K 中．设 $f(\alpha)$ 是在 K 中的那一个 x 或 y，如果它们都在 K 中则任选其中一个．注意，因为 M 是一个匹配，所以如果 α 和 β 是 M 的两条边，那么 $f(\alpha)$ 必定不同于 $f(\beta)$（为什么）．因此，我们可以为 M 的每一条边指定 K 中的一个不同元素．因此，M 的元素个数一定不超过 K．

因此，如果 M^* 是最大匹配，K^* 是最小覆盖，则有 $|M^*| \leqslant |K^*|$．注意，$|M^*|$ 可以严格小于 $|K^*|$．在图 12.13 的图中，最小覆盖是由顶点 1，3，5，6 组成的，所以 $|K^*|=4$．然而，一个最大匹配是由三条边组成的．例如，$M^*=\{\{1, 2\}, \{3, 4\}, \{5, 6\}\}$ 是一个最大匹配．因此，$|M^*|=3<|K^*|$．下面是一个重要结果．

702

定理 12.4　*如果 M 是匹配，而 K 是覆盖，且 $|M|=|K|$，则 K 是最小覆盖而 M 是最大匹配．*

证明　如果 M^* 是任意最大匹配，而 K^* 是任意最小覆盖，那么我们有

$$|M| \leqslant |M^*| \leqslant |K^*| \leqslant |K|. \tag{12.3}$$

因为 $|M|=|K|$，所以(12.3)式中的所有项必须相等，特别地，有 $|M|=|M^*|$ 和 $|K|=|K^*|$．■

例如，考虑图 12.9 中的图．这里，存在由四条边 $\{S_1, a\}\{S_3, b\}$，$\{S_4, c\}\{S_6, d\}$ 组成的匹配，而四个顶点 a，b，c，d 组成一个覆盖 K．因此，根据定理 12.4，M 是最大匹配，K 是最小覆盖．这个例子说明下面的定理．

定理 12.5(König[1931])　*假设 $G=(X, Y, E)$ 是一个二部图．那么最大匹配的边数等于最小覆盖中的顶点数量．*

这一定理的证明在 13.3.8 节．

作为这一定理的推论，现在我们可以得到 Philip Hall 定理，定理 12.1．

Philip Hall 定理的证明(定理 12.1)[⊖]　我们还需证明：如果对于所有的 $S \subseteq X$，有 $|N(S)| \geqslant |S|$，那么存在使 X 饱和的匹配．根据定理 12.5，因为 G 是二部图，所以使 X 饱和的匹配存在当且仅当对于每一个覆盖 K 有 $|K| \geqslant |X|$．设 K 是一个覆盖，且设 $X-S$ 是 K 中的 X 顶点的集合．这时，S 的顶点不在 K 中，所以 $N(S)$ 的所有顶点一定都在 K 中．所以下式成立：

$$|K| \geqslant |X-S|+|N(S)|=|X|-|S|+|N(S)| \geqslant |X|.$$

因此，对于所有的 K 有 $|K| \geqslant |X|$，于是存在使 X 饱和的匹配．■

定理 12.5 的另一个推论对于 12.7 节中的优化分配问题的算法很重要（练习 12.6）．为了说明这一推论，设 $\mathbf{A}$ 是 0 和 1 构成的矩阵．这一矩阵的一条线(line)或者是一行或者是一列．这个矩阵中的 0 的一个集合称为是*独立的*(independent)，如果没有两个 0 位于同一条线上．例如，假设有如下矩阵 $\mathbf{A}$：

$$\mathbf{A}=\begin{bmatrix}1 & 0 & 0 & 1\\ 0 & 1 & 0 & 0\\ 1 & 0 & 0 & 1\\ 1 & 0 & 1 & 1\end{bmatrix}, \tag{12.4}$$

这时，项(2，1)和(1，3)形成一个 $\mathbf{A}$ 的 0 独立集合，而(2，1)，(4，2)，(3，3)也形成一个 $\mathbf{A}$ 的 0 独立集合．

703

⊖ 本定理证明可以跳过．

推论 12.5.1(König-Egervary 定理[1])　*如果 $\boldsymbol{A}$ 是 0 和 1 构成的矩阵，那么最大 0 独立集合的元素数量等于 $\boldsymbol{A}$ 中覆盖所有 0 的线的最小集合的元素数量.*

证明　用下面的方法构建一个二部图 $G=(X, Y, E)$：X 是 $\boldsymbol{A}$ 的行，Y 是 $\boldsymbol{A}$ 的列，且由一条边连接行 i 和行 j，当且仅当 $\boldsymbol{A}$ 的 i，j 项是 0，那么 $\boldsymbol{A}$ 的最大 0 独立集合对应于 G 的最大匹配，而 $\boldsymbol{A}$ 中覆盖所有 0 的线的最小集合对应于 G 的最小覆盖. ■

为了说明这一结果，对于(12.4)式中的 $\boldsymbol{A}$，最大 0 独立集合是位于项(2，1)，(4，2)，(3，3)处的三个 0 的集合. $\boldsymbol{A}$ 中覆盖所有 0 的线的最小集合是由第二行、第二列和第三列组成的.

12.4.2　边覆盖

一个图的边的集合 F 称为*边覆盖*(edge covering)，如果这个图的每一个顶点都在 F 中的一条边上. 一个边覆盖称为是*最小的*(minimum)，如果没有其他边数更少的边覆盖. 最小边覆盖被应用于计算机工程学中的开关函数、晶体物理学以及其他领域(参见 Deo[1974，pp. 184ff.]). 在练习中，我们研究边覆盖及其与匹配及顶点覆盖间的关系.

本节练习

1. 对于图 12.14 中的图 a，寻找：
   (a)五个顶点的覆盖　　(b)最小覆盖
2. 对于图 12.1 中的图，寻找：
   (a)六个顶点的覆盖　　(b)最小覆盖
3. 一家公司在试图改进它的计算机操作的安全性的过程中插入了 20 个特殊密码. 每一个密码这家公司中正好有两个人知道. 寻找同时知道所有密码的人的最小集合. 使用本节的术语阐述这一问题.
4. 假设 $\boldsymbol{A}$ 是 0 和 1 构成的矩阵. 假设我们寻找覆盖 $\boldsymbol{A}$ 中所有 0 的 k 条线，并且我们发现了 k 个独立 0. 我们能够得出什么结论?
5. 如果允许矩阵以任意整数为项，König-Egervary 定理还为真吗? 为什么?
6. 对于下面每一个图，寻找一个不是最小的边覆盖和一个最小的边覆盖.
   (a)图 12.13 中的图　　(b)图 12.14 中的图 a
   (c)图 12.14 中的图 b　　(d)图 12.12 中的图 b

704

7. (a)有孤立顶点的图能有边覆盖吗?
   (b)在有 n 个顶点的图的边覆盖中，最小可能边数是多少?
8. 一辆巡逻车停在一个街区的中央，它可以观察这个街区端处的两个交叉路口. 你如何确定使附近的所有街角都在监视之下所需的巡逻车的最小数量(参见 Gondran 和 Minoux[1984])?
9. (Minieka[1978])一个委员会是按下面的方法选举而成的：50 个州的每个州至少有一名成员，而美国 65 个主要少数民族中的每一个民族至少有一名成员. 如果这一要求得到满足，那么由一组志愿者组成的最小委员会是什么? 为了回答这一问题，设一个图的顶点是 50 个州和 65 个少数民族，并设每名志愿者对应于连接他所在州和所在少数民族的边. 我们要寻找什么来回答这个问题?
10. 通过计算下面各图中的最大匹配和最小覆盖来说明关于栅格图 $G_{n1,n2}$(由 11.2.4 节定义)的定理 12.5：
    (a)$G_{1,2}$　　(b)$G_{1,m}$　　(c)$G_{2,2}$　　(d)$G_{n,n}$
11. (a)证明定理 12.5 至少对 11.2.5 节所定义的某些环形栅格 AN(c，s)不成立.
    (b)存在使它成立的任意环形栅格吗?
12. (a)一个图的最小边覆盖能包含回路吗? 为什么?
    (b)证明有 n 个顶点的图的每一个最小边覆盖至多有 $n-1$ 条边.
13. (Gallai[1959])假设 G 是一个没有孤立顶点的图，K^* 是 G 的最小(顶点)覆盖，而 I^* 是顶点的最大独立

[1] 本定理基于 König[1931]和 Egervary[1931]的工作.

集合. 证明 $|K^*|+|I^*|=|V|$.

14. (Norman 和 Rabin[1959]和 Gallai[1959])假设 G 是一个没有孤立顶点的图，M^* 是最大匹配，而 F^* 是最小边覆盖. 证明 $|M^*|+|F^*|=|V|$.
15. 假设 I 是一个没有孤立顶点的图 G 的顶点的独立集合，而 F 是 G 的一个边覆盖. 证明 $|I|\leqslant|F|$.
16. 假设我们要寻找一个没有孤立顶点的图 G 的顶点独立集合 I 和 G 的一个边覆盖 F 使得 $|I|=|F|$. 我们能得出什么结论? 为什么?
17. 假设 $G=(X, Y, E)$ 是一个没有孤立顶点的二部图. 设 I^* 是一个有最大顶点数量的独立集合(**最大独立集合**(maximum independent set))，而 F^* 是 G 的最小边覆盖. 通过练习 13 和练习 14 证明 $|I^*|=|F^*|$.
18. (Minieka[1978])考虑没有孤立顶点的图的如下算法. **算法 1** 从一个匹配 M 开始，并选择任意不饱和顶点 x. 它把属于 x 的任意边加入到 M 中，并反复这一过程直到每一个顶点都是饱和的. 最终的边集合记作 C'. **算法 2** 开始于一个边覆盖 C，并选择由 C 中的多条边覆盖的任意顶点 x. 它从 C 中删除覆盖 x 的一条边，并反复这一过程直到没有顶点被多条边覆盖. 最终的边集合记作 M'.

705

   (a)证明算法 1 产生的 C' 是一个边覆盖.
   (b)证明算法 2 产生的 M' 是一个匹配.
   (c)证明如果 M 是最大匹配，那么 C' 是最小边覆盖.
   (d)证明如果 C 是最小边覆盖，那么 M' 是最大匹配.
19. 从定理 12.4 推导 Philip Hall 定理(定理 12.1).

12.5 寻找最大匹配

12.5.1 M 增广链

本节给出寻找一个图 G 的最大匹配的过程. 假设 M 是 G 的某个匹配. G 中的一条 **M 交互链**(M-alternating chain)是一条简单链:

$$u_1, e_1, u_2, e_2, \cdots, u_t, e_t, u_{t+1}, \tag{12.5}$$

使得 e_1 不在 M 内，e_2 在 M 内，e_3 不在 M 内，e_4 在 M 内，以此类推. 例如，考虑图 12.15 中由波线表示的匹配. 这时，链 a, $\{a, b\}$, b, $\{b, g\}$, g, $\{g, f\}$, f, $\{f, e\}$, e 是一条 M 交互链. 如果一个 M 交互链(12.5)连接两个非 M 饱和的顶点 u_1 和 u_{t+1}(即 u_1 和 u_{t+1} 不在 M 的任意边上)，我们称这条链是一条 **M 增广链**(M-augmenting chain). 我们的例子不是 M 增广链，因为 e 是 M 饱和的. 然而下面的链是 M 增广链:

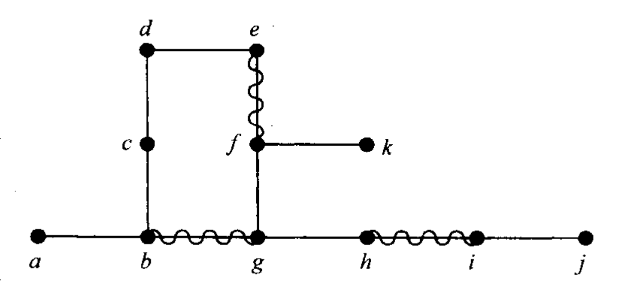


图 12.15 波线边表示一个匹配 M

$$a,\{a,b\},b,\{b,g\},g,\{g,f\},f,\{f,e\},e,\{e,d\},d. \tag{12.6}$$

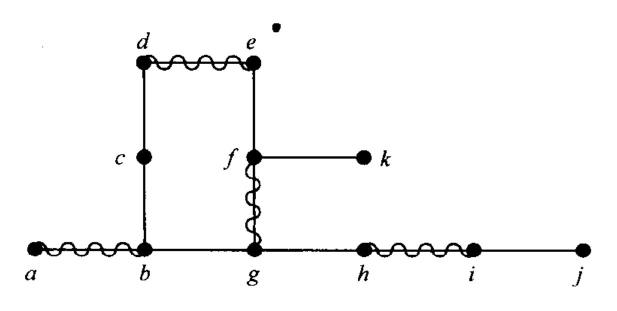


图 12.16 通过使用 M 增广链(12.6)由图 12.15 的匹配 M 得到的匹配 M'

现在，我们通过从 M 中删除所有用在 M 增广链(12.6)中的 M 的边，并加入增广链(12.6)中不在 M 中的所有边，来寻找一个新匹配 M'. 于是，M' 如图 12.16 所示. 它的确是一个匹配. 另外，M' 的边数比 M 多一条. 这一过程总给出更大的匹配.

定理 12.6　*假设 M 是一个匹配，C 是 M 的一个 M 增广链．设 M' 是从 M 中减去在 M 中的 C 的边，加上不在 M 中的 C 的边．这时，M' 是一个匹配，且 $|M'|>|M|$．*　706

证明　为了证明 $|M'|>|M|$，注意在 M 增广链(12.5)中，e_1 和 e_t 不能在 M 内，所以 t 是奇数．因此，在加入边 e_1，e_3，…，e_t 并删除边 e_2，e_4，…，e_{t-1} 时，我们加入的边数比我们拿走的边数多一条．为了弄明白 M' 是一个匹配，注意如果 j 是奇数，那么边 e_j 被加入，而边 e_{j+1} 被减去．因此，如果 j 是奇数且 $j\neq 1$，$t+1$，那么之前 u_j 只能在 e_{j-1} 上，而现在它只能在 e_j 上．如果 j 是偶数，那么之前 u_j 在 e_j 上而现在它只在 e_{j-1} 上．同样，之前 u_1 没有被匹配，现在只能在 e_1 上，而 u_{t+1} 之前没有被匹配，现在只能在 e_t 上．　■

作为定理 12.6 的一个推论，我们观察到，如果 M 有 M 增广链，那么 M 不是最大匹配．事实上，逆叙也成立．

定理 12.7(Berge[1957]、Norman 和 Rabin[1959])　*G 的匹配 M 是最大的当且仅当 G 不包含 M 增广链．*

为了运用这一定理，注意图 12.6 中的匹配 M' 不是最大的，因为存在如下的 M' 增广链：

$$j,\{j,i\},i,\{i,h\},h,\{h,g\},g,\{g,f\},f,\{f,k\},k. \tag{12.7}$$

如果我们通过删除链(12.7)的 M' 中的边 $\{i,h\}$ 和 $\{g,f\}$，并加入链(12.7)的非 M' 的边 $\{j,i\}$，$\{h,g\}$ 和 $\{f,k\}$，我们得到图 12.17 所示的匹配 M''．不存在 M'' 增广链，因为只有一个未饱和顶点．因此，M'' 是最大的．

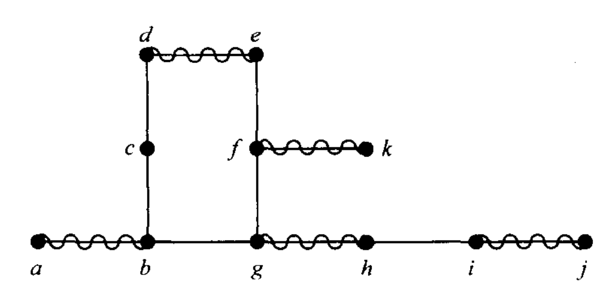


图 12.17　根据图 12.6 的匹配 M' 通过使用 M' 增广链(12.7)得到的匹配 M''

12.5.2　定理 12.7 的证明[⊖]

为了证明定理 12.7，还需证明：如果存在匹配 M' 使得 $|M'|>|M|$，那么存在 M 增广链．设 H 是 G 的子图，它的顶点由 G 的所有顶点组成，而它的边由 G 的所有在 M 中或在 M' 中，但不是既在 M 又在 M' 中的边组成．注意，在 H 中，M' 中的边数大于 M 的边数．另外，H 的每个顶点在 H 中至多有两个邻居，因为它在 M 中至多有一个邻居，在 M' 中至多有一个邻居．利用后者的观察结果，可以证明(参见练习 10)H 的每一个连通分支或者是一个回路 Z_n，或者是有 n 个顶点的简单链 L_n．另外，Z_n 或 L_n 中的边在 M 和 M' 之间必须是交互的，因为 M 和 M' 是匹配．因此，每一个 Z_n 取自 M 的边和取自 M' 的边的数量相同．由此，因为 H 中 M' 的边数大于 M 的边数，形如 L_n 的某个分支具有这一性质．这个 L_n 一定是形如(12.5)式的链，且 e_1，e_3，…，e_t 在 M' 中而不在 M 中，而 e_2，e_4，…，e_{t-1} 在 M 中而不在 M' 中．另外，u_1 和 u_{t+1} 不能是 M 饱和的，因为否则 L_n 不是 H 的分支．因此，链 L_n 是 M 增广链．这就完成了 12.7 的证明．　707

为了说明这一证明，考虑图 12.18 所示的匹配 M 和 M'．这时，图 H 也如图中所示．链

$$g,\{g,h\},h,\{h,i\},i,\{i,j\},j$$

是包含 M' 的两条边和 M 的一条边的简单链．　708

⊖　本小节可以跳过．

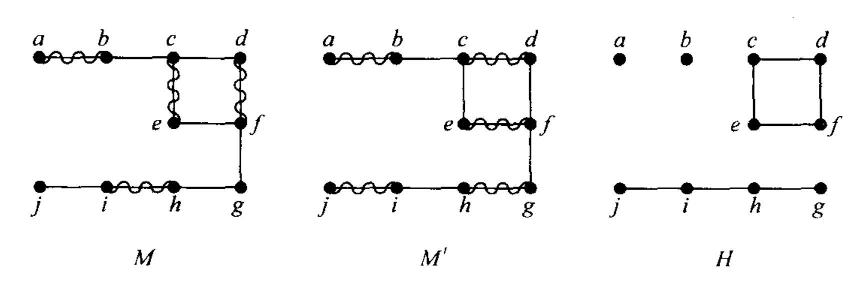


图 12.18　从匹配 M 和 M' 得到的图 H

12.5.3　寻找最大匹配的算法

接下来，我们描述一个寻找最大匹配的算法. 这一算法源自 L. R. Ford 和 D. R. Fulkerson 的工作(参见 Ford 和 Fulerson[1962]). 这一算法有两个基本子程序. 子程序 1 搜索开始于一个不饱和顶点 x 的 M 增广链，如果子程序 1 寻找到 M 增广链，则子程序 2 构建一个更大的匹配 M'. 算法选择一个不饱和顶点 x 并运用子程序 1. 如果发现一个开始于 x 的 M 增广链，那么调用子程序 2. 如果没有发现这样的链，那么在子程序 1 中使用另一个不饱和顶点 y. 这一过程将反复进行，直到或者发现一个 M 增广链或者没有剩余不饱和顶点. 对于后者的情况，我们得出不存在 M 增广链，所以有一个最大匹配的结论.

子程序 2 的运行正如定理 12.6 所描述的那样. 现在我们给出子程序 1. 描述这一子程序的最容易方法就是使用一个二部图 G 来说明. 如果 G 不是二部图，那么这一过程将会更复杂一些. 正是 Edmonds[1965a]首次观察到的那样，这里需要一些更巧妙的东西(参见练习 16). 关于二部和非二部匹配的更详细论述，参见 Ahuja，Magnanti 和 Orlin[1993]、Cook 等[1998]、Lawler[1976]、Lovász 和 Plummer[1986]，Minieka[1978]、Papadimitriou 和 Steiglitz[1982].

子程序 1 开始于一个匹配 M 和 M 中的一个不饱和顶点 x，而且分阶段构建一棵称为*交互树*(alternating tree)的树 T. 其想法是 x 在 T 中，且 T 中开始于 x 的所有简单链是 M 交互链. 例如，对于图 12.15，如果 $x=a$，这样的一棵交互树是由下面的边组成的，$\{a, b\}$，$\{b, g\}$，$\{g, h\}$，$\{g, f\}$. 交互树 T 上的顶点称为外部顶点或内部顶点. 顶点 y 称为*外部的*(outer)，如果 x 和 y 之间的唯一简单链[⊖]结束于 M 中的边. 否则顶点 y 是*内部的*(inner). 顶点 x 称为是*外部的*(outer). 如果发现一棵带有一个不饱和顶点 $y\neq x$ 的交互树 T，那么 x 和 y 之间的唯一简单链是一个 M 增广链. 把顶点和边加入到 T，直到或者发现这样的一个链，或者不再有可以加入到 T 中的顶点. 对于后者的情况，不存在 M 增广链. 现在我们详细给出这个子程序.

算法 12.2：子程序 1——搜索一个开始于顶点 x 的 M 增广链

输入：一个二部图 G，G 的一个匹配 M，以及 M 中的一个不饱和顶点 x.

输出：一个开始于 x 的 M 增广链，或不存在这样的链的信息.

步骤 1. 设置 $T=\varnothing$，$T'=\varnothing$(T 是树中的边的集合，T' 是肯定不在树中的边的集合). 集合 709 $O=\{x\}$，$I=\varnothing$(O 是外部顶点的集合，I 是内部顶点的集合).

步骤 2.

步骤 2.1. 在既不在 T 中也不在 T' 中的各边中，如果不存在外部顶点(O 中的顶点)和其他任意顶点之间的边，那么前进到步骤 3. 否则设 $\{u, v\}$ 是 $u\in O$ 的这样的一条边.

步骤 2.2. 如果顶点 v 是一个内顶点(在 I 中)，那么把边 $\{u, v\}$ 放入 T' 中并重复步骤 2.1.

⊖ 我们要使用定理 3.18 的结果，即在一棵树中，任意一对顶点之间存在唯一简单链.

如果顶点 v 既不是外部顶点也不是内部顶点，那么把边$\{u, v\}$放入 T 中，并前进到步骤 2.3(因为 G 是二部图，v 不可能是外部顶点．因为否则你可以证明 G 有奇回路，根据定理 3.4 对于二部图这是不可能的，参见练习 11).

步骤 2.3. 如果 v 不是饱和的，停止．T 中从 x 到 v 的唯一链形成从 x 到 v 的 M 增广链．如果 v 是饱和的，在 M 中存在唯一的边$\{v, w\}$．这时，把边$\{v, w\}$放入 T 中，v 在 I 中且 w 在 O 中．返回步骤 2.1.

步骤 3. 只有当向 T 或 T'进一步加入边是不可能的时候，我们进入到这一步骤．停止并给出不存在开始于 x 的 M 增广链的信息.

注意，这个算法以两种方式停止，发现了一条 M 增广链，或发现了一棵交互树 T，其中已不能把边加入到 T 或 T'中．对于前者的情况，过程前进到子程序 2．对于后者的情况，可以证明不存在从顶点 x 开始的 M 增广链．我们对另一个不饱和顶点 x 重复这一子程序．如果在重复使用子程序 1 时，我们没有找到从不饱和顶点 x 开始的 M 增广链，那么我们得出这个匹配是最大匹配的结论.

我们对于图 12.19 中的匹配 M 说明这一算法．挑选一个不饱和顶点 x 为 a，并称 a 为外部顶点．这是步骤 1．前进到步骤 2.1，选择边$\{a, b\}$，它既不在 T 内也不在 T' 内，且连接外部顶点．因为 b 不是外部也不是内部的，在步骤 2.2 中我们把这条边放入到 T 中，并前进到步骤 2.3．因为 b 是饱和的，所以我们考虑 M 的唯一边$\{b, g\}$．我们把这条边放入到 T 中，并称 b 是内部的，而称 g 是外部的，并返回到步骤 2.1(参见表 12.3 的概括，为了这一概括的目的，考虑返回到步骤 2.1 的迭代).

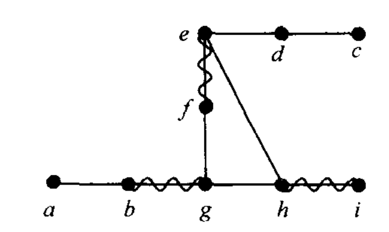


图 12.19　波形线表示一个匹配 M

表 12.3　运用于图 12.19 中的 M 的算法 12.2 的各步骤，且开始于顶点 $x=a$

迭代	T	T'	I(内部顶点)	O(外部顶点)
1	$\varnothing$	$\varnothing$	$\varnothing$	a
2	$\{a, b\}$	$\varnothing$	$\varnothing$	a
	$\{a, b\}, \{b, g\}$	$\varnothing$	b	a, g
3	$\{a, b\}, \{b, g\}, \{g, f\}$	$\varnothing$	b	a, g
	$\{a, b\}, \{b, g\}, \{g, f\}, \{f, e\}$	$\varnothing$	b, f	a, g, e
4	$\{a, b\}, \{b, g\}, \{g, f\}, \{f, e\}, \{g, h\}$	$\varnothing$	b, f	a, g, e
	$\{a, b\}, \{b, g\}, \{g, f\}, \{f, e\}, \{g, h\}, \{h, i\}$	$\varnothing$	b, f, h	a, g, e, i
5	$\{a, b\}, \{b, g\}, \{g, f\}, \{f, e\}, \{g, h\}, \{h, i\}$	$\{e, h\}$	b, f, h	a, g, e, i
6	$\{a, b\}, \{b, g\}, \{g, f\}, \{f, e\}, \{g, h\}, \{h, i\}, \{e, d\}$	$\{e, h\}$	b, f, h	a, g, e, i

在步骤 2.1 中，我们考虑既不在 T 中也不在 T'中，且连接一个外部顶点的边．有两条这样的边$\{g, h\}$，$\{g, f\}$．假设我们选择了边$\{g, f\}$．因为 f 既不是外部的也不是内部的，所以在步骤 2.2，我们把$\{g, f\}$放入 T 中．现在 f 是饱和的，所以我们考虑 M 的唯一边$\{f, e\}$．我们把这条边放入 T 中，称 f 为内部的，而 e 为外部的，并返回到步骤 2.1.

接下来，我们再一次考虑既不在 T 中也不在 T'中，且连接一个外部顶点的边．这样的可能边有$\{g, h\}$，$\{e, h\}$和$\{e, d\}$．假设我们选择$\{g, h\}$．那么在步骤 2.2 中，因为 h 既不是内部的也不是外部的，所以我们把边$\{g, h\}$放入 T 中．于是在步骤 2.3 中，因为 H 是饱和的，所以我们把边$\{h, i\}$放入 T 中，并称 h 为内部的，i 为外部的.

我们再一次考虑既不在 T 中也不在 T'中，且连接一个外部顶点的边．这样的可能边有$\{e,$

$h\}$和$\{e, d\}$. 假设我们选择了前者. 那么顶点 h 是内部的，所以在步骤 2.2 中，我们把边$\{e, h\}$放入 T'中，并重复步骤 2.1.

在步骤 2.1 中，现在我们选择边$\{e, d\}$，而且在步骤 2.2 中我们把边$\{e, d\}$加入 T 中. 于是在步骤 2.3 中，因为 d 不是饱和的，我们停止，并寻找 T 中的唯一从 x 到 v 的链，即从 a 到 d 的链. 这样的链是 a, $\{a, b\}$, b, $\{b, g\}$, g, $\{g, f\}$, f, $\{f, e\}$, e, $\{e, d\}$, d. 这是一条 M 增广链.

在结束之际，我们注意到，可以证明，给定一个二部图 $G=(X, Y, E)$，我们可以修改所描述的算法，使其需要$[\min\{|X|, |Y|\}]\cdot|E|$步骤量级. 因此，这一算法是一个关于顶点数量 $n=|V|$ 的多项式算法. 因为 $|E|\leqslant\binom{n}{2}\leqslant n^2$ 且 $\min\{|X|, |Y|\}\leqslant n$. 因此，这一算法以 n^3 步骤量级结束. 用 2.18 节的记法，我们说这一算法是 $O(n^3)$算法. 对于任意图 G 的相关算法也大约花费 n^3 步骤量级. 到我们写这本书的时候，已知最快的寻找最大匹配的算法花费$|E|\,|V|^{1/2}$步骤量级. 用 n 表示，这些算法大约花费 $n^{5/2}$ 步骤量级. 关于寻找最大匹配的算法的计算复杂度，对于二部图的情况，要归功于 Hopcroft 和 Karp[1973]，而对一般情况，归功于 Micali 和 Vazirani[1980]. 这些算法把匹配与网络流联系起来(我们将 13.3.8 节讨论这一关系). 有关于匹配算法的计算复杂度的详细讨论，参见 Ahuja，Magnanti 和 Orlin[1993]、Cook 等[1998]、Lawler[1976]、Papadimitrionu 和 Steiglitz[1982]. 710~711

本节练习

1. 对于图 12.20 中的匹配：
   (a)寻找一个不是 M 增广链的 M 交互链.
   (b)如果存在 M 增广链，寻找一个这样的链.
   (c)使用(b)的链，如果存在的话，寻找一个更大的匹配.
2. 对于图 12.21 中的匹配重复练习 1.
3. 对于图 12.22 中的匹配重复练习 1.
4. 对于图 12.23 中的匹配重复练习 1.
5. 对于图 12.24 中的匹配重复练习 1.

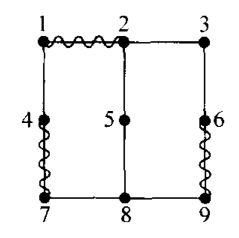


图 12.20　12.5 节练习的匹配

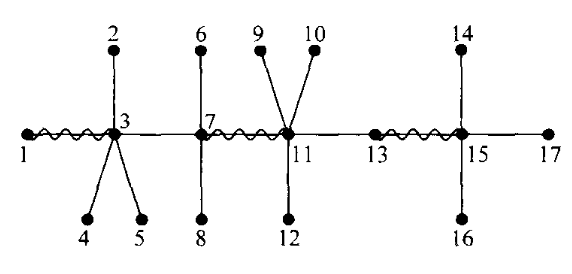


图 12.21　12.5 节练习的匹配

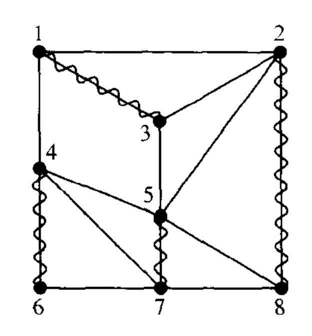


图 12.22　12.5 节练习的匹配

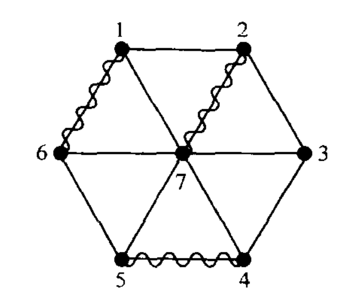


图 12.23　12.5 节练习的匹配

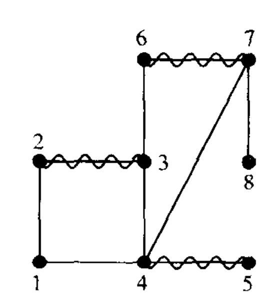


图 12.24　12.5 节练习的匹配

6. 把子程序 1 运用于下面的情况：
   (a)图 12.20 中的匹配 M 和顶点 5
   (b)图 12.20 中的匹配 M 和顶点 3
   (c)图 12.21 中的匹配 M 和顶点 12
7. 把子程序 1 运用于图 12.24 中匹配 M，以顶点 1 开始，且在之后的迭代中，分别选择 $v=2$，$v=4$，$v=6$，$v=4$，$v=8$.
8. 证明：如果把子程序 1 运用于图 12.23 中的匹配，开始于顶点 3，有可能在步骤 2.1 处选择边$\{u, v\}$来连接两个外部顶点. 为什么会发生这样的情况?
9. 对于下列情况，寻找开始于顶点 x 的交互树.
   (a)$x=6$ 和图 12.21 中的匹配
   (b)$x=8$ 和图 12.20 中的匹配
10. 证明在定理 12.7 的证明中，H 的每一个分支是 Z_n 或 L_n.
11. 证明在算法 12.2 的步骤 2.2 中，v 不可能是外部顶点.
12. 当运用子程序 2 时，证明一旦一个顶点被饱和，那么它就保持被饱和.
13. 证明算法 12.2 可行.
14. 如果顶点 x 是匹配M 中的不饱和顶点，且不存在开始于 x 的M 增广链，证明存在其中 x 是不饱和的最大基数匹配.
15. 顶点的子集 S 称为*可匹配的*(matchable)，如果 S 中每一个顶点在某个匹配中被匹配. 证明：如果 S 是可匹配的，那么 S 中每一个顶点在某个最大基数匹配中被匹配.
16. 假设 M 是图G 的一个匹配. 一个相对于 M 的*花簇*(blossom)是一个有 $2k+1$ 个顶点且在 M 中有 k 条边的奇数长度回路B. 证明：如果不存在相对于 M 的花簇，那么如果存在开始于 x 的 M 增广链的话，子程序 1 可以找到一个这样的 M 增广链(因此，仅当找到花簇时，我们才必须修改所描述的寻找最大匹配的算法. Edmonds[1965a]所做的修改是通过搜索花簇并把它们缩减成单一顶点，然后在结果图中寻找M增广链完成的).

712~713

12.6 匹配尽可能多 X 的元素

假设 $G=(X, Y, E)$是二部图. 如果 G 中不存在使 X 饱和的匹配，那么我们至少可以希望寻找一个匹配，它匹配尽可能多 X 的元素. 设 $m(G)$是 G 的匹配中可以被匹配的 X 元素的最大数量. 我们说明如何计算 $m(G)$. 首先，我们给出这一想法的几个应用.

【例 12.17 电话切换网络】 在电话切换站，电话呼叫从呼入线被发送到呼出线. *切换网络*(switching network)把每一条呼入线与某些呼出线连接起来. 一个呼叫沿着一条呼入线进来，它通过切换网络到达一条呼出线. 切换网络可以用二部图 $G=(X, Y, E)$表示. X 的顶点是呼入线，Y 的顶点是呼出线，呼入线和呼出线之间有一条边，当且仅当前者与后者是连通的. 假设一次进入多个呼叫. 我们想在同一时间内尽可能把它们发送出去. 如果 S 是有呼叫的呼入线的集合，我们想要给 S 的每一条线指定一条导向不同呼出线的G 的边. 即我们想要一个 G 的 S 饱和匹配. 当不可能这样做时，我们就要匹配 S 中较大数量的线. 一般地，我们想要设计一个切换网络，使得如果呼叫进入到所有可能的呼入线，那么我们能够把它们中的大部分与呼出线匹配. 即我们设计一个切换网络，使得它有很多边的最大匹配. 换句话说，对于给定的 X 和 Y，我们要寻找一个二部图 $G=(X, Y, E)$使得 $m(G)$合理地足够大. ■

【例 12.18 天花疫苗接种(再探例 12.3)】 在提前制定的公众健康紧急事件，如天花的处理计划中，我们想要确定有足够的诊所给人们接种. 然而，接种诊所也许没有能力给每一个人接种，至少在短时间内不能给每一个人接种. 假设每一个接种诊所每一天至多给特定数量的人接

714

种. 我们想要安排人与诊所之间的分配，使得我们能够在第一天尽可能给更多的人接种. 换句话说，我们想要做人与接种诊所间的分配，定义一个图 G，使得 $m(G)$尽可能大. ■

如果 $S\subseteq X$，设 S 的亏度(deficiency)$\delta(S)$定义为 $|S|-|N(S)|$，并设 G 的亏度(deficiency)$\delta(G)$是 $\max_{S\subseteq X}\delta(S)$. 注意，$\delta(\varnothing)=0$，所以，根据 Philip Hall 定理，$G$ 有使 X 饱和的匹配当且仅当 $\delta(G)=0$.

定理 12.8(König[1931]) 如果 $G=(X, Y, E)$是二部图，那么 $m(G)=|X|-\delta(G)$.

为了说明此定理，考虑图 12.25 中的图. 注意，$\delta(\{a, b, d\})=2$，且这是最大的，所以$\delta(G)=2$. 因此，$m(G)=|X|-\delta(G)=4-2=2$. 可以匹配的 X 的最大子集有两个元素. 这样的集合的一个例子是$\{a, c\}$.

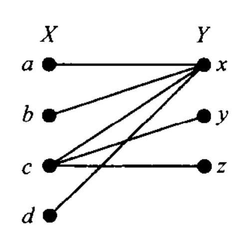


图 12.25 有 $\delta(G)=2$ 的图 G

练习 6 和练习 7 概述了定理 12.8 的证明.

本节练习

1. 对于图 12.12 中的每一个图，计算 $\delta(G)$和 $m(G)$.
2. 考虑有 9 条呼入线和 6 条呼出干线的切换站. 假设根据工程学的考虑，每一条呼入线在这一切换网络中正好与两条呼出线连通，而且每一条呼出线至多与 3 条呼入线连通.
   (a)考虑 p 条呼入线的任意集合 S. 证明在这一切换网络中，S 中的顶点的度数总和至多等于 $N(S)$中的顶点的度数总和.
   (b)使用(a)的结果证明，对于有 p 条呼入线的任意集合 S，$N(S)$至少有$\frac{2}{3}p$ 个元素.
   (c)对于有 p 条呼入线的任意集合 S，得出 $\delta(S)\leqslant\frac{1}{3}p$ 的结论.
   715 (d)证明：无论切换网络是如何设计的，总存在一个匹配，它至少匹配 6 条呼入呼叫.
3. 假设我们修改练习 2 的情况，使得有 12 条呼入线和 10 条呼出线. 如果每一条呼入线正好与 3 条呼出线连通，而每一条呼出线至多与 4 条呼入线连通，证明无论这一切换网络是如何设计的，总是存在一个匹配，它至少匹配 9 条呼入呼叫.
4. 假设我们修改练习 2 的情况，使得有两种类型的呼入线，类型Ⅰ有 4 条呼入线，类型Ⅱ也有 4 条呼入线，且有 8 条呼出线，它们有相同类型. 假设类型Ⅰ的每一条呼入线正好与 3 条呼出干线连通，而类型Ⅱ的每一条呼入线正好与 6 条呼出线连通，并假设每一条呼出线至多与 4 条呼入线连通. 证明无论这一切换网络是如何设计的，总是存在一种匹配，它至少匹配 7 条呼入呼叫. 考虑类型Ⅰ的 p 条呼入线和类型Ⅱ的 q 条呼入线的集合 S，做上述相同的工作. 证明有：
$$\delta(S)\leqslant(p+q)-\frac{3p+6q}{4}.$$
   得出因为 $p\leqslant 4$ 且 $q\geqslant 0$，所以 $\delta(G)\leqslant 1$ 的结论.
5. 假设在一个小城镇里，有 4 个天花疫苗接种诊所和 56 个人. 另外，每一个人将在他们所选择的两个接种诊所中的一个完成接种，但是每一个诊所的日承受能力为 14 人. 定义一个二部图 G，使得$m(G)$对应于这个小镇中一天所接种的最大人数.
6. 作为定理 12.8 的证明的准备，证明下面的事实：二部图 $G=(X, Y, E)$有 k 个元素的匹配，当且仅当对于所有子集 $S\subseteq X$，$|N(S)|\geqslant|S|+k-|X|=k-|X-S|$[提示：通过向 Y 中加入 $|X|-k$个顶点，并把 Y 中的每一个新顶点与 X 中的每一个顶点相连来修改 G].
7. 使用练习 6 的结果证明定理 12.8.
8. 本练习给出 König-Egerváry 定理(推论 12.5.1)的另一个证明. 设 $\boldsymbol{A}$ 是由 0 和 1 构成的矩阵，如 König-Egerváry 定理的证明那样，根据 $\boldsymbol{A}$ 构建一个二部图 $G=(X, Y, E)$. 设 S 是 $\boldsymbol{A}$ 的行集合 X 的一个子集，使得 $\delta(S)=\delta(G)$.

(a)证明：对应于 $X-S$ 的顶点的行以及对应于 $N(S)$ 的顶点的列包含 $\mathbf{A}$ 中的所有 0.

(b)证明：(a)中的所有行和列的数量总和等于 $|X|-\delta(G)$.

(c)证明：覆盖所有 0 的 $\mathbf{A}$ 的线的最小数量至多是 $|X|-\delta(G)$.

(d)证明：G 中的独立 0 的最大数量是 $|X|-\delta(G)$.

(e)证明 König-Egerváry 定理.

12.7 最大权匹配

在 12.1 节的例 12.5～例 12.9 中，我们介绍了寻找在边上有权的图的最大权匹配问题. 这里，我们更详细地讨论这一问题，并用这些问题的术语给出解决这一问题的算法. 716

12.7.1 再论"中国邮差"问题

在 11.4.1 节，我们讨论了希望寻找能够覆盖指定路线上所有街区且行走最少街区的邮差问题. 我们如下阐述这一问题. 给定一个图 G，我们研究可行多重图(feasible multigraph) H，这个多重图有一条封闭的欧拉链，且这个图是根据 G 通过增加 G 的边的拷贝而得到的. 我们寻找一个最优多重图(optimal multigraph)，即有最小边数的可行多重图. 在 11.4.2 节中，我们发现在计算机绘图领域也出现这样的问题. 在 11.4 节的练习 11 中，要求读者注意，在任意最优多重图 H 中，新增加的边可以被划分成连接图 G 中奇度数顶点的链，其中任意一个这样的顶点正好是一个这样链的一个端顶点. 假设对于 G 中的奇度数的每一对顶点 u 和 v，我们寻找 G 中它们之间的最短链[我们可以使用类似于 Dijkstra 算法(13.2.2 节)的算法去做这件事]. 我们通过取图 G 中的所有奇度数顶点作为顶点来构建一个图 G'，用一条边连接每一对顶点，并给这条边设置一个等于 u 和 v 之间的最短链长度的权. 为了说明，假设 G 是图 11.28 中的图，图 12.26 重新给出这个图. 那么 G 中有 4 个奇度数顶点：b，f，h，d. 这 4 个顶点中的每一对顶点之间，显然最短链的长度都是 2. 因此，图 G' 如图 12.26 中第二部分所示. 现在，由 G 而得的任意最优多重图 H 对应于连接 G 中的奇度数顶点对的链的集合，其中每一个这样的顶点正好为这样的链的一个端端点. 这样的链集合定义图 G' 中的一个完美匹配. 而且，一个最优图 H 对应于这样的链的一个集合，使得这样的链的长度之和尽可能小. 这一集合本身又定义 G' 中最小权的完美匹配. 如果通过对每个权取负来修改 G'，那么我们要寻找 G' 的最大权匹配(这样的匹配也是完美的，为什么). 这是我们在例 12.5～例 12.9 中所讨论的主要问题. 对于我们的例子，G' 的最小权完美匹配是由任意两条不邻接的边组成的，例如 $\{b, d\}$ 和 $\{f, h\}$. 这时，对应于所选择的边的最短链将定义一个最优多重图 H. 这里，b 和 d 之间的最短链是 b，e，d，而 f 和 h 之间的最短链是 f，x，h. 如果我们把这些链上的边加到 G 中，我们得到图 11.28 中的第一个最优多重图 H. 现在我们知道为了得到一个有欧拉闭链的 H，这是可以加入的最小边数. H 中的这条欧拉闭链给出邮递员在 G 中的一条最优路径. 717

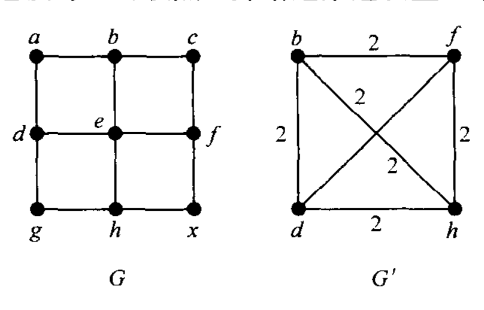


图 12.26 图 11.28 中的图 G 和对应的图 G'

我们所描述的方法归功于 Edmonds(参见 Edmonds[1965b]、Edmonds 和 Johnson[1973]). 应该指出的是，我们所描述的过程是一个高效的过程. 因为高效的最短路径算法(13.2.2 节的 Dijkstra 算法)可以在 n^2 步骤量级内完成，而且可以在 n^3 步骤量级内完成最小(最大)权匹配(Lawler[1976]).

12.7.2 最优分配问题(最大权匹配)的算法[⊖]

在例12.6中，我们讨论了下面的工作分配问题. 有 n 名工人和 m 项工作，每名工人适合每一项工作，已知工人 i 对工作 j 的可能能力由等级 r_{ij} 给出. 我们希望给工人们指定工作，使得可以最大化能力等级的和. 这是一个最大权匹配问题的例子. 我们假设 $m=n$，并设 x_{ij} 是这样的变量：如果工人 i 被指定工作 j 它等于1，否则它等于0. 于是，因为 $m=n$，所以我们要最大化下式：

$$\sum_{\substack{i=1\\j=1}}^{\substack{m\\n}} r_{ij}x_{ij}.$$

我们要求没有两个工人得到相同的工作，即

$$\sum_{i=1}^{n} x_{ij} \leqslant 1. \tag{12.8}$$

我们还要求每名工人都得到一项工作，即

$$\sum_{j=1}^{n} x_{ij} \geqslant 1. \tag{12.9}$$

于是，因为工人的数量等于工作的数量，所以(12.8)式和(12.9)式给出下面的限制条件：

$$\sum_{i=1}^{n} x_{ij} = 1 \quad 和 \quad \sum_{j=1}^{n} x_{ij} = 1. \tag{12.10}$$

现在我们给出一个解决最优分配问题的算法. 假设我们有一个 $n\times n$ 矩阵 (c_{ij})，而且我们希望寻找数 $x_{ij}=0$ 或 $x_{ij}=1$，使得我们可以最小化 $\sum_{ij} c_{ij}x_{ij}$，而且(12.10)式成立. 注意，如果从 (c_{ij}) 的第 k 行的所有项中减去一个常量 p_k，给出另一个矩阵 (c'_{ij})，那么根据(12.10)式有

718

$$\sum_{i,j} c'_{ij}x_{ij} = \sum_{i,j} c_{ij}x_{ij} - p_k\sum_{j} x_{kj} = \sum_{i,j} c_{ij}x_{ij} - p_k,$$

因此，最小化 $\sum_{ij} c_{ij}x_{ij}$ 的分配 x_{ij} 也最小化 $\sum_{ij} c'_{ij}x_{ij}$. 如果从 (c_{ij}) 的第 l 列的每一项中减去一个常量 q_l，结果是类似的. 实际上，对于所有 i，从 (c_{ij}) 的第 i 行的所有项中减去常量 p_i，而对于所有的 j，从 (c_{ij}) 的第 j 列的所有项中减去常量 q_j 也有相同的结果.

使用上述观察结果，我们设 p_i 是 (c_{ij}) 的第 i 行中的最小元素. 对于每一个 i，我们把 (c_{ij}) 的第 i 行的每一个元素都减去 p_i，得到一个矩阵 (c'_{ij}). 现在我们设 q_j 是 (c'_{ij}) 的第 j 列中的最小元素，对于每一个 j，我们把 (c'_{ij}) 的第 j 列的每一个元素减去 q_j，得到一个矩阵 $(\overline{c}_{ij})$. 我们把这一矩阵称为简化矩阵(reduced matrix). 根据我们上面所做的观察，我们可以使用这个简化矩阵来解决最优分配问题.

为了说明这一过程，假设 $n=4$ 且矩阵 (c_{ij}) 为

$$(c_{ij}) = \begin{bmatrix} 12 & 14 & 15 & 14 \\ 9 & 6 & 11 & 8 \\ 10 & 9 & 16 & 14 \\ 12 & 13 & 13 & 10 \end{bmatrix}. \tag{12.11}$$

这时，$p_1=12$，$p_2=6$，$p_3=9$，$p_4=10$ 且矩阵 (c'_{ij}) 为

$$(c'_{ij}) = \begin{bmatrix} 0 & 2 & 3 & 2 \\ 3 & 0 & 5 & 2 \\ 1 & 0 & 7 & 5 \\ 2 & 3 & 3 & 0 \end{bmatrix}.$$

⊖ 本小节的内容可以推迟到13.4节后阅读.

现在，$q_1=0$，$q_2=0$，$q_3=3$，$q_4=0$，所以矩阵(c_{ij})为

$$(\overline{c_{ij}})=\begin{bmatrix}0&2&0&2\\3&0&2&2\\1&0&4&5\\2&3&0&0\end{bmatrix}. \qquad (12.12)$$

注意简化矩阵$(\overline{c_{ij}})$总有非负项(为什么). 给每名工人分配一项工作且每一项工作只有一名工人的工作分配x_{ij}对应于这一简化矩阵的n个项的选择，即每一行上有一个项且每一列上有一个项. 当且仅当我们选择i，j项时有$x_{ij}=1$. 现在，假设我们可以确定n个项的一种选择，其中每一行上有一个项且每一列上有一个项，使得所有项都等于0. 为了明确这n个i，j项，我们取$x_{ij}=1$，否则取$x_{ij}=0$. 那么，对应的$\sum_{ij}\overline{c_{ij}}x_{ij}$将等于0，因为当$x_{ij}=1$时，$\overline{c_{ij}}=0$. 例如，如果矩阵$(\overline{c_{ij}})$为

$$(\overline{c_{ij}})=\begin{bmatrix}3&0&1\\0&5&4\\0&3&0\end{bmatrix},$$

719

我们可以取1，2项，2，1项和3，3项. 如果我们取$x_{12}=1$，$x_{21}=1$，$x_{33}=1$，否则$x_{ij}=0$，那么$\sum\overline{c_{ij}}x_{ij}=0$. 显然这是最小的，因为由于$\overline{c_{ij}}\geqslant 0$，我们有$\sum\overline{c_{ij}}x_{ij}\geqslant 0$.

回想一下，在12.4.1节，一个矩阵的0独立集合是这样的0的集合：其中任意两个0不在同一行也不在同一列上. 我们刚刚观察到的结果是，如果我们能够在$(\overline{c_{ij}})$中找到一个0独立集合，那么我们就能够通过取对应的项x_{ij}等于1，而找到一个最优的工作分配. 因为在$(\overline{c_{ij}})$中不存在超过n个0的0独立集合，所以我们寻找最大的0独立集合. 我们又将如何寻找这样的集合呢？让我们把矩阵$(\overline{c_{ij}})$的所有正项都改成1. 于是，因为$\overline{c_{ij}}$是非负的，所以其结果矩阵是0和1构成的矩阵. 回想一下，在12.4.1节，一个矩阵的线(line)或者是一行或者是一列. 所以，根据König-Egerváry定理(推论12.5.1)，$(\overline{c_{ij}})$中独立0的最大数量，等价地，在其由0和1构成的修改矩阵中独立0的最大数量，等于覆盖所有0的线的最小数量. 因此，我们可以通过检查是否存在覆盖所有0的n条线的集合，来检查是否存在n个独立0的集合. 另一方面，根据König-Egerváry定理的证明，回想一下矩阵$(\overline{c_{ij}})$中的0的最大独立集合，等价地，修改后的0和1矩阵的0的最大独立集合，对应于二部图$G=(X, Y, E)$中的最大匹配，其中$X=Y=\{1, 2, \cdots, n\}$，而且在此图中，当且仅当$\overline{c_{ij}}=0$时，X中的i和Y中的j之间存在边. 因此，我们可以运用12.5.3节的最大匹配算法来验证是否存在n个独立0的集合.

在等式(12.11)的例子中，我们在等式(12.12)得到$(\overline{c_{ij}})$. 注意，覆盖$(\overline{c_{ij}})$中的所有0的线的最小数量是3：使用第一行，第四行和第二列(参见图12.27). 因此，0的最大独立集合有三个元素，这不足以给出我们所要求的工作分配，因为我们需要四个独立0.

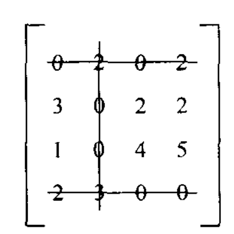


图12.27 0的最小线覆盖

解决最优分配问题的算法是通过持续修改简化矩阵$(\overline{c_{ij}})$进行的，使得最终得到可以找到n个独立0的简化矩阵. 修改的步骤是使用线的0的最小覆盖，诸如图12.27所示的图，并寻找最小的未覆盖元素. 然后，从每一个未覆盖元素减去这个最小元素，并把它加到每一个两次覆盖元素上，寻找一个新的矩阵，在这个新矩阵中搜索独立0. 对于我们的例子，最小未覆盖元素是1，它是处于3，1的元素. 我们把矩阵$(\overline{c_{ij}})$的2，1项，2，3项，2，4项，3，1项，3，3项和3，4项减去1，并把它加入到1，2项和4，2项，得到如下新简化矩阵：

720

$$\begin{bmatrix} 0 & 3 & 0 & 2 \\ 2 & 0 & 1 & 1 \\ 0 & 0 & 3 & 4 \\ 2 & 4 & 0 & 0 \end{bmatrix}. \tag{12.13}$$

不难证明，可以在前面的简化矩阵的不同行或不同列上加上或减去一个常量而得到这个新的简化矩阵(练习 8)．因此，使用这一矩阵来解决最优分配问题与使用前面的简化矩阵来解决最优分配问题是一样的．而且，这个新的简化矩阵的所有项都是非负的．因此，我们可以使用这一新矩阵，再一次寻找有 n 个 0 的独立集合．我们通过寻找线的最小覆盖来寻找这样的集合．在(12.13)式中，最小覆盖使用四条线，因此，一定存在四个 0 的独立集合．一个这样的集合由 1，3 项，2，2 项，3，1 项和 4，4 项组成．因此，我们通过设 $x_{13}=1$，$x_{22}=1$，$x_{31}=1$，$x_{44}=1$，而所有其他 $x_{ij}=0$ 来得到一个最优分配．如果我们没有找到有 n 个 0 的独立集合，那么我们要重复修改简化矩阵的过程．

我们所描述的算法称为*匈牙利算法*，它归功于 Kuhn[1955]．我们将其概括如下．

算法 12.3：最优分配问题的匈牙利算法

输入： 一个 $n\times n$ 矩阵 (c_{ij})．

输出： 一个最优分配 x_{ij}．

步骤 1. (初始化)

步骤 1.1. 对于每个 i，设 p_i 是矩阵 (c_{ij}) 的第 i 行中的最小元素．

步骤 1.2. 设 (c'_{ij}) 是对于所有 i，通过把矩阵 (c_{ij}) 的第 i 行中的每个元素减去 p_i 而得的矩阵．

步骤 1.3. 对于每一个 j，设 q_j 是 (c'_{ij}) 的第 j 列中的最小元素．

步骤 1.4. 设 $(\overline{c_{ij}})$ 是对于所有 j，把矩阵 (c'_{ij}) 的第 j 列的每个元素减去 q_j 而得的矩阵．

步骤 2.

步骤 2.1. 寻找覆盖 $(\overline{c_{ij}})$ 的 0 的线的最小集合．

步骤 2.2. 如果这个线的集合的元素数量小于 n，进入步骤 3，否则进入步骤 4．

步骤 3. (修改简化矩阵)

[721]

步骤 3.1. 使用在步骤 2.1 得到的覆盖，设 p 是 $(\overline{c_{ij}})$ 中的最小未覆元素．

步骤 3.2. 通过从每一个未覆盖元素中减去 p，并把 p 加到每个两次覆盖元素来修改简化矩阵 $(\overline{c_{ij}})$．返回到步骤 2．

步骤 4.

步骤 4.1. 寻找 $(\overline{c_{ij}})$ 中的 n 个 0 的独立集合．

步骤 4.2. 如果 $(\overline{c_{ij}})$ 的 i，j 项是独立 0 之一，则设 x_{ij} 等于 1，否则设 x_{ij} 等于 0．输出这个解 x_{ij} 并停止．

定理 12.9 *如果所有的 c_{ij} 都是整数，那么匈牙利算法给出一个最优分配．*

证明㊀ 我们已经观察到，如果这一算法给出一个分配 x_{ij}，那么它一定是最优的．但是，我们如何知道这一算法将会给出一个分配呢？这一算法会给出一个分配的理由是：因为简化矩阵总是有非负整数项，而且因为在每一次修改时，正如我们下面所看到的那样，这一矩阵的所有项的和减少一个至少等于 1 的整数．因此，经过有限步骤，如果我们还没有得到一个最优解的话，这个简化矩阵的所有项都将等于 0．对于这种情况，当然存在 n 个独立 0 的集合，所以我们

㊀ 本证明可以跳过．

可以找到一个最优解.

现在我们证明，如果$(\bar{d}_{ij})$是从简化矩阵$(\bar{c}_{ij})$通过步骤 3 而得到简化矩阵，那么

$$\sum_{i,j}(\bar{c}_{ij})-\sum_{i,j}(\bar{d}_{ij})$$

是一个不小于 1 的整数.

回想一下，我们有一个$k<n$条线的$(\bar{c}_{ij})$的覆盖. 设S_r是$(\bar{c}_{ij})$的未覆盖行的集合，S_c是未覆盖列的集合，$\bar{S}_r$是覆盖行的集合，$\bar{S}_c$是覆盖列的集合. 设$\alpha=|S_r|$且$\beta=|S_c|$. 那么$k=(n-\alpha)+(n-\beta)$. 同时，回想一下，p是$(\bar{c}_{ij})$的最小未覆盖项. 于是，由我们从$(\bar{c}_{ij})$得到$(\bar{d}_{ij})$的方法，我们有

$$\begin{aligned}\sum_{i,j}\bar{c}_{ij}-\sum_{i,j}\bar{d}_{ij}&=\sum_{\substack{i\in S_r\\ j\in S_c}}[\bar{c}_{ij}-\bar{d}_{ij}]+\sum_{\substack{i\in S_r\\ j\in \bar{S}_c}}[\bar{c}_{ij}-\bar{d}_{ij}]+\sum_{\substack{i\in \bar{S}_r\\ j\in S_c}}[\bar{c}_{ij}-\bar{d}_{ij}]+\sum_{\substack{i\in \bar{S}_r\\ j\in \bar{S}_c}}[\bar{c}_{ij}-\bar{d}_{ij}]\\&=\sum_{\substack{i\in S_r\\ j\in S_c}}p+\sum_{\substack{i\in S_r\\ j\in \bar{S}_c}}0+\sum_{\substack{i\in \bar{S}_r\\ j\in S_c}}0+\sum_{\substack{i\in \bar{S}_r\\ j\in \bar{S}_c}}(-p)\\&=\alpha\beta p-(n-\alpha)(n-\beta)p\\&=n(\alpha+\beta-n)p.\end{aligned}$$

而$\alpha+\beta$是未覆盖行和列的数量，所以

$$\alpha+\beta-n=(2n-k)-n=n-k>0,$$

因为$k=(n-\alpha)+(n-\beta)$，且$k<n$. 因此，因为n，$\alpha+\beta-n$和p都是正整数，所以

$$\sum_{i,j}\bar{c}_{ij}-\sum_{i,j}\bar{d}_{ij}=n(\alpha+\beta-n)p$$

是一个不小于 1 的整数. ■

722

最后，如果有n名工人和n项工作，使用 2.18 节的术语，匈牙利算法可以在$O(n^3)$的时间内完成. 关于这一点的讨论，参见 Papadimitriou 和 Steiglitz[1982].

本节练习

1. 对于图 11.39 中的每一个图G，如 12.7.1 节中对“中国邮差”问题的解所描述的那样，寻找对应的图G'. 寻找G'中的最小权完美匹配并把这一匹配翻译成“中国邮差”问题的一个解.
2. 下面的每一个矩阵给出使用工人i做工作j的成本c_{ij}. 使用匈牙利算法寻找最小成本工作分配.

(a)
$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4\\ \hline 1 & 8 & 3 & 2 & 4\\ 2 & 10 & 9 & 3 & 6\\ 3 & 2 & 1 & 1 & 5\\ 4 & 3 & 8 & 2 & 1\end{array}$$

(b)
$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4\\ \hline 1 & 17 & 5 & 8 & 11\\ 2 & 3 & 9 & 2 & 10\\ 3 & 4 & 2 & 8 & 6\\ 4 & 7 & 6 & 4 & 5\end{array}$$

(c)
$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5\\ \hline 1 & 8 & 7 & 5 & 11 & 4\\ 2 & 9 & 7 & 6 & 11 & 3\\ 3 & 12 & 9 & 4 & 8 & 2\\ 4 & 1 & 2 & 3 & 5 & 6\\ 5 & 11 & 4 & 2 & 8 & 2\end{array}$$

3. 下面的矩阵给出工人i做工作j的等级r_{ij}. 使用匈牙利算法寻找一个最优(即最高效率)的工作分配.

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5\\ \hline 1 & 8 & 7 & 5 & 9 & 6\\ 2 & 11 & 9 & 7 & 4 & 8\\ 3 & 12 & 6 & 7 & 5 & 10\\ 4 & 9 & 7 & 6 & 9 & 6\\ 5 & 3 & 9 & 8 & 9 & 8\end{array}$$

4. 一家公司购买了不同类型的 5 台新机器. 在它的工厂，有 5 个可以放置这些机器的位置. 对于给定的机器，放置的位置对处理材料的难易程度有影响. 例如，如果这台机器靠近为它生产原料的工作中心，那

么它将是高效的. 对于位置j处的机器i, 运送给这台机器的原料处理的小时成本可以计算出来. 下面的矩阵给出这一信息.

$$
\text{机器}\quad
\begin{array}{c}
\\ 1\\2\\3\\4\\5
\end{array}
\overset{\text{位置}}{
\begin{array}{c}
\begin{array}{ccccc}1&2&3&4&5\end{array}\\
\begin{bmatrix}
4&6&8&5&7\\
2&4&6&9&5\\
1&7&6&8&3\\
4&6&7&5&7\\
10&4&5&4&5
\end{bmatrix}
\end{array}}
$$

723

我们如何给机器分配位置, 使得可以最小化最终的原料处理成本呢?

5. 下面的矩阵给出扬声器i和j配对时的频率响应f_{ij}(参见练习12.8).

(a)使用匈牙利算法寻找最小频率响应配对.

$$
\begin{array}{c}
\\ 1\\2\\3\\4\\5\\6
\end{array}
\begin{array}{c}
\begin{array}{cccccc}1&2&3&4&5&6\end{array}\\
\begin{bmatrix}
0&4&8&8&7&8\\
4&0&6&8&1&6\\
8&6&0&5&7&4\\
8&8&5&0&4&2\\
7&1&7&4&0&7\\
8&6&4&2&7&0
\end{bmatrix}
\end{array}
$$

(b)如果某一行上的所有项相同(对角线上的项除外, 它们必须是0), 那么这将如何影响匈牙利算法的使用?

6. 回想一下, 12.1节例12.5的房地产交易例子. 假设一个代理商有三套房子H_1, H_2和H_3, 且有四位买主B_1, B_2, B_3, B_4. 如果B_1买房子H_1, H_2或H_3, 那么她的报价分别是20, 18和8千美元, 如果B_2买一套房子, 她的报价分别是20, 12和16千美元, 如果B_3买一套房子, 报价分别是18, 20和20千美元, 如果B_4买一套房子, 报价分别是16, 18和16千美元. 对于她的最大利润问题, 是否存在唯一解? [提示: 引入"虚设的"第四套房子.]

7. (a)假设在12.1节例12.7的天花接种问题中, 必须把某些人分配到特定的接种诊所. 解释如何满足这一要求. 特别地, 需要对匈牙利算法的输入矩阵做什么样的修改?

(b)现在假设在12.1节例12.7的天花接种问题中, 某些人不能分配到特定的接种诊所. 解释如何满足这一要求. 特别地, 需要对匈牙利算法的输入矩阵做什么样的修改?

8. 证明: 在匈牙利算法中, 新简化矩阵可以通过把旧简化矩阵的不同行或列减去或加上一个常量而得到.

12.8 稳定匹配

在例12.12中, 我们介绍了如何分配医学院毕业生到某些医院作实习医生的问题. 1952年, 人们开发出一个称为全国实习医生匹配程序的算法来做这样的分配. 其思想是实习医生应该提交他们申请的医院的等级, 而且医院将类似地等级化它们的申请者. 现在称为全国居民匹配程序的算法本质上没有太大的改变且还在使用. 其目的在于产生一个"稳定的"匹配.

把优先选择与匹配结合起来产生一个稳定匹配的思想是由Gale和Shapley[1962]提出的. 我们在4.2.5节粗略地讨论了这一思想. 这一工作还带来理论上的巨大进展以及更进一步的实际应用. 关于这一方面的深入讨论, 参见Gusfield和Irving[1989]、Roth和Sotomayor[1990]. (作为稳定性的另一个方法, Balinski和Ratier[1997]使用有向图的术语重新开发了稳定匹配的思想.)

724

由Gale和Shapley[1962]提出的稳定匹配的一个特殊类型解决了稳定婚姻问题. **大小为n的稳定婚姻问题**(stable marriage problem of size n)涉及n名男士和n名女士的集合. 每个人提出一

个对于所有异性成员的(严格)优先选择顺序. 如果为每个人建立起一夫一妻制的婚姻, 那么这一婚姻集合称为一个匹配, 即男士和女士之间的一对一的对应关系. 一个匹配称为是**不稳定的**(unstable), 如果某名男士和某名女士, 按照他们的优先选择顺序, 愿意离开他们各自的配偶(相互结合). 这样的一对男女称为**中断对**(blocking pair). 如果不存在中断对, 那么这一匹配是稳定的.

本节考虑与稳定婚姻问题相关的问题中的存在问题、计数问题和优化问题. 下面的例子突出强调这三点.

【例 12.19 稳定婚姻】㊀ 假设有四名男士和四名女士要相互结婚. 他们的优先选择顺序由表 12.4 给出. 我们用 m_i-w_j 表示 m_i 和 w_j 之间的婚姻. 注意 $M=\{m_1-w_4, m_2-w_3, m_3-w_1, m_4-w_2\}$ 是一个婚姻的稳定集合. 为了弄明白其原因, 注意, w_4, w_3 分别与他们的第一选择结婚; 所以双方都不愿意离开他们的配偶 m_1 和 m_2. 同样, m_3 和 m_4 在其余的女士, 即 w_1 和 w_2 中获得了他们的最好选择.

表 12.4 大小为 4 的稳定婚姻问题的优先选择顺序

男士的优先选择				女士的优先选择			
m_1	m_2	m_3	m_4	w_1	w_2	w_3	w_4
w_1	w_2	w_3	w_4	m_4	m_3	m_2	m_1
w_2	w_1	w_4	w_3	m_3	m_4	m_1	m_2
w_3	w_4	w_1	w_2	m_2	m_1	m_4	m_3
w_4	w_3	w_2	w_1	m_1	m_2	m_3	m_4

虽然 M 是稳定的, 但实际上还存在另外 9 种与这些人和他们的优先选择相关的稳定匹配. 总共是 $4!=24$ 种匹配(为什么). $M_u=\{m_1-w_1, m_2-w_3, m_3-w_4, m_4-w_2\}$ 是不稳定匹配的一个例子. 为了弄明白原因, 注意 m_2 尽管已与 w_3 结婚, 但实际上他更喜欢 w_4(还有 w_1 和 w_2). w_4 虽已与 m_3 结婚, 但实际上她更喜欢 m_2(和 m_1). 因为 m_2 和 w_4 互相喜欢对方胜过喜欢各自的配偶, 所以他们形成一个中断对, 因此, M_u 是不稳定的.

另外两个明显的稳定匹配是 $M_m=\{m_1-w_1, m_2-w_2, m_3-w_3, m_4-w_4\}$ 和 $M_w=\{m_1-w_4, m_2-w_3, m_3-w_2, m_4-w_1\}$. 这两个匹配都是稳定的, 因为 M_m 中的每名男士及 M_w 中的每名女士都得到了他们的第一选择. 同样, 因为这一事实, 注意, 与其他任意稳定匹配相比, 所有男士都更喜欢 M_m, 而所有女士都更喜欢 M_w. ■ 725

12.8.1 Gale-Shapley 算法

每一个稳定婚姻问题都存在婚姻的稳定集合吗? Gale 和 Shapley[1962]对这一存在问题给出了肯定的答案. 我们描述寻找稳定匹配的 Gale-Shapley(GS)算法. GS 算法考虑了临时"订婚"直到最后的阶段, 即当所有的订婚终结于婚姻时. 一个阶段有两个步骤, 其中所有自由男士(未订婚)求婚, 然后所有被求婚的女士都订婚.

GS 算法开始于每名男士向他的优先选择列表上的 1 号女士求婚. 然后每名女士与她的优先选择列表中排在最高位的求婚者订婚. 在步骤 2, 所有剩余的自由男士向他们的 2 号女士求婚, 然后每一位女士与她的新的求婚者, 或她的未婚夫中最中意者订婚, 如果存在这样的选择的话(注意在任意阶段, 有可能有些女士不能做出任何决定, 因为在那个阶段没有人向她们求婚).

每一个后继阶段都以相同的方式进行. 所有自由男士向他的优先选择列表上他还没有求婚的最喜欢的女士求婚, 每一位女士把她的新求婚者和未婚夫比较以获得她的最好选择, 然后步

㊀ 本例来自于 Gusfield 和 Irving[1989].

入她的新婚约. 当所有女士(男士)都已订婚时，GS算法终止.

GS算法所产生的匹配是稳定的吗? 假设 m_i 和 w_j 没有结婚，但是 m_i 较之他的妻子 w_k 更喜欢 w_j. 那么一定是 m_i 在向 w_k 求婚之前先向 w_j 求婚. 然而，当 w_j 接到 m_i 的求婚时，或者她拒绝他而接受她所选择的某人，或与 m_i 订婚而后来拒绝他并接受了一个更好的求婚者. 无论是哪种情况，较之 m_i，w_j 更喜欢她的丈夫，因而由GS算法所产生的匹配是稳定的.

注意，GS算法也可以以女士求婚的方式进行操作，并产生一个稳定匹配. 这一稳定匹配可能不同于用"男士求婚"方式得到的匹配.

为了计算GS算法的复杂度，我们注意到，在每一步，接到求婚的女士或者(i)第一次接受求婚，或者(ii)拒绝新求婚或解除婚约. 对于(i)她正好做一次，而对于(ii)她至多做 $n-1$ 次. 因此，对每名女士这个算法有 $O(n)$ 步骤，因此总共有 $O(n^2)$ 步骤.

对于GS算法做最后的说明. 我们通过所有自由男士向他优先选择列表上的下一位喜欢的女士求婚来描述一个阶段；然而，并没有给出男士应该这样做的顺序. Gusfield 和 Irving[1989]已证明特殊的顺序不影响GS算法的结果.

定理 12.10(Gusfield 和 Irving[1989]) *GS算法的所有可能执行都产生相同的稳定匹配.*

726 为了弄明白GS算法的使用，考虑表12.5中的四位男士和四位女士的优先选择列表. GS算法开始于 m_1，m_2，m_3，m_4 分别向 w_1，w_2，w_3，w_4 求婚. 这些求婚所产生的婚约是 m_1-w_1，m_3-w_3，m_4-w_2(注意 w_2 选择 m_4，因为在她优先选择列表中 m_4 高于 m_2). 第二阶段 m_2 向他的第二选择 w_1 求婚. 她接受他的求婚，并解除与 m_1 的婚约，因为在她的优先选择列表中 m_2 高于 m_1. 第三阶段是 m_1 向他的第二选择 w_4 向求婚. 她接受他的求婚，而且他们订婚. 此时，所有女士(和男士)都已订婚，所以GS算法以所有婚约变成(稳定)婚姻而停止.

表 12.5 大小为 4 的稳定婚姻问题的优先选择顺序

男士的优先选择				女士的优先选择			
m_1	m_2	m_3	m_4	w_1	w_2	w_3	w_4
w_1	w_2	w_3	w_2	m_4	m_3	m_2	m_1
w_4	w_1	w_4	w_3	m_3	m_4	m_1	m_2
w_3	w_4	w_1	w_4	m_2	m_1	m_4	m_3
w_2	w_3	w_2	w_1	m_1	m_2	m_3	m_4

尽管GS算法保证至少存在一个稳定匹配，但很多稳定婚姻问题正好只存在一个稳定匹配，即GS算法产生唯一的稳定匹配(练习6). 在下一节，我们考虑可能是大小为 n 的所有稳定婚姻问题中稳定匹配的最大数量.

12.8.2 稳定匹配的数量

在大小为2的任意稳定婚姻问题中，总共存在 $2!=2$ 个匹配. 根据GS算法，总是至少存在一个稳定匹配. 接下来的例子说明所有的(两个)匹配都可能是稳定的. 考虑大小为2且有表12.6的优先选择列表的稳定婚姻问题. 容易验证，$M_m=\{m_1-w_1, m_2-w_2\}$ 和 $M_w=\{m_1-w_2, m_2-w_1\}$ 都是稳定的. 注意，M_m 是男士求婚时GS算法的输出，而 M_w 是女士求婚时GS算法的输出.

表 12.6 大小为 2 的稳定婚姻问题的优先选择顺序

男士的优先选择		女士的优先选择	
m_1	m_2	w_1	w_2
w_1	w_2	m_2	m_1
w_2	w_1	m_1	m_2

727 在例12.19中，我们说总共24种匹配中这一问题存在10种稳定匹配. 对于较小的例子，不难枚举出所有的匹配并一一验证每一个是否是稳定的. (练习7要求读者设计一个验证稳定匹配

的算法）．然而，对于较大的问题，枚举是不可能的，因为大小为 n 的稳定婚姻问题有 $n!$ 种匹配．下面的定理给出大小为 n 的问题的稳定婚姻集合数量的下界．（寻找大小为 n 的稳定婚姻集合的最大可能数量是一个未解问题，参见 Knuth[1997]．）

定理 12.11（Irving 和 Leather[1986]） *对于每一个 $n\geqslant 1$，n 是 2 的幂，存在大小为 n 且至少有 2^{n-1} 个稳定匹配的稳定婚姻问题．*

证明 对 n 做归纳来证明这一定理．对于 $n=1=2^0$，$2^{n-1}=2^0=1$，根据 GS 算法，在任意稳定婚姻问题中，包括大小为 1 的婚姻问题，一定存在（至少）一个稳定匹配．对于 $n=2=2^1$，表 12.6 描述的问题给出大小为 2 且有 $2=2^{2-1}$ 个稳定匹配的稳定婚姻问题．这一稳定婚姻问题的例子对此证明非常重要．假设这一定理对 $n=2^k$ 成立．即假设存在大小为 $n=2^k$ 且至少有 $2^{n-1}=2^{2k-1}$ 个稳定匹配的稳定婚姻问题．假设在这种情况下男士和女士分别被标记为 b_1，b_2，…，b_{2^k} 和 g_1，g_2，…，g_{2^k}．

回想一下，m_1，m_2，和 w_1，w_2 分别是表 12.6 中的大小为 2 的稳定婚姻问题中的男士和女士．为了完成这一证明，我们需要创建一个大小为 2^{k+1} 且至少有 $2^{2^{k+1}-1}$ 个稳定匹配的新稳定婚姻问题．

考虑大小为 2^{k+1} 且有下面的男士和女士的稳定匹配问题．

$$\text{男士}:(b_1,m_1),\cdots,(b_{2^k},m_1),(b_1,m_2),\cdots,(b_{2^k},m_2)$$

和

$$\text{女士}:(g_1,w_1),\cdots,(g_{2^k},w_1),(g_1,w_2),\cdots,(g_{2^k},w_2).$$

上述的每名男士的优先选择列表如下所示：

$(b_i,\ m_j)$ 较之 $(g_{k'},\ w_{l'})$ 更喜欢 $(g_k,\ w_l)$，如果 m_j 较之 $w_{l'}$ 更喜欢 w_l，或者如果 $l=l'$ 且 b_i 较之 $g_{k'}$ 更喜欢 g_k．

女士的优先选择列表也做类似的定义．现在，只要证明这一问题至少存在 $2^{2^{k+1}-1}$ 个稳定的匹配即可．

设 M_1，M_2，…，M_n 是表 12.6 的稳定婚姻问题的任意（不必是不同的）稳定匹配序列；注意，每个 M_i 有两种选择．设 M 是上面所述的大小为 $n=2^k$ 问题的（至少）2^{2^k-1} 个稳定匹配中的任意一个．因此，对于这 $n+1$ 个稳定匹配 M_1，M_2，…，M_n，可能选择的总数 M 至少是 $2\cdot 2\cdot 2\cdots 2\cdot 2^{2^k-1}$，这个数等于 $2^n\cdot 2^{2^k-1}=2^{2^k}\cdot 2^{2^k-1}=2^{2^{k+1}-1}$．接下来，我们要证明在大小为 2^{k+1} 的新稳定婚姻问题中，不同的稳定匹配可以通过这些组合的任意一个来定义．这意味着对于大小为 2^{k+1} 的稳定婚姻问题，我们已发现至少 $2^{2^{k+1}-1}$ 个稳定的匹配，于是就可以完成证明． 728

根据稳定匹配序列 M_1，M_2，…，M_n，M，我们如下定义一个婚姻：

$$(b_i,m_j)\text{ 与}(M\text{ 中 }b_i\text{ 的配偶},M_i\text{ 中 }m_j\text{ 的配偶})\text{ 结婚．}\tag{12.14}$$

只要证明这些婚姻形成一个匹配，每一个匹配都是不同的且匹配是稳定的，就可完成这一证明．这三个事实的证明留给读者作为练习． ■

其中，定理 12.11 证明了稳定婚姻问题的稳定匹配的数量随着问题的大小而成指数增加．⊖ 尽管寻找给定稳定婚姻问题的所有稳定匹配，无论是在算法上还是在数量上都很困难，但是我们仍然可以评定稳定匹配的结构．这就是我们下一节所要做的事情．

12.8.3 稳定匹配的结构

正如 4.2.5 节中那样，我们可以如下定义稳定匹配的*支配*（dominance）关系：一个人 x 喜欢一个稳定匹配 M_i 超过另一个 M_j，如果与 M_j 中的他的配偶相比，x 更喜欢 M_i 中的他的配偶．*面向男士支配关系*（man-oriented dominance relation）是说，M_i 支配 M_j，如果每一位男士或者更

⊖ Irving 和 Leather[1986]发现了大幅度改进定理 12.11 的递推关系．

喜欢 M_i 而不是 M_j，或者对二者的喜好没有差异(回想一下，你对选择 A 和 B 没有差异，如果你既不是较之 B 更喜欢 A，也不是较之 A 更喜欢 B). *面向女士支配关系*(woman-oriented dominance relation)的定义类似.

正如我们在 4.2.5 节中所描述的那样，稳定匹配在面向男士支配关系或面向女士支配关系下，形成一个偏顺(实际上，它们形成一个格). (这一事实的证明是练习 9 的一部分). 考虑与表 12.4 的稳定婚姻问题相关的所有稳定匹配. 有如下 10 种匹配：

$$
\begin{aligned}
M_0 &= \{m_1 - w_4, m_2 - w_3, m_3 - w_2, m_4 - w_1\} \\
M_1 &= \{m_1 - w_3, m_2 - w_4, m_3 - w_2, m_4 - w_1\} \\
M_2 &= \{m_1 - w_4, m_2 - w_3, m_3 - w_1, m_4 - w_2\} \\
M_3 &= \{m_1 - w_3, m_2 - w_4, m_3 - w_1, m_4 - w_2\} \\
M_4 &= \{m_1 - w_3, m_2 - w_1, m_3 - w_4, m_4 - w_2\} \\
M_5 &= \{m_1 - w_2, m_2 - w_4, m_3 - w_1, m_4 - w_3\} \\
M_6 &= \{m_1 - w_2, m_2 - w_1, m_3 - w_4, m_4 - w_3\} \\
M_7 &= \{m_1 - w_1, m_2 - w_2, m_3 - w_4, m_4 - w_3\} \\
M_8 &= \{m_1 - w_2, m_2 - w_1, m_3 - w_3, m_4 - w_4\} \\
M_9 &= \{m_1 - w_1, m_2 - w_2, m_3 - w_3, m_4 - w_4\}.
\end{aligned}
\tag{12.15}
$$

729

在面向男士支配关系之下，本例的格图示如图 12.28 所示. 格的性质之一是它们有最大元素和最小元素. 面向男士格中的最大稳定匹配和最小稳定匹配分别称为*男士最优*(man-optimal)和*男士最劣*(man-pessimal). 这是因为，面向男士支配的定义形式，在面向男士格的最大元素中，每名男士的结婚对象都是在任意稳定匹配中与其结婚的所有女士中等级最高的女士. 然而，注意，当考虑该格中的所有稳定匹配时，w_1 与 m_1，m_2，m_3 或 m_4 结婚. 在这四名男士中，她最不喜欢 m_1，而他就是在面向男士格的最大元素中与她结婚的男士. 对于所有女士情况都是这样；每一位女士在面向男士格的最大元素中的结婚对象，都是在任意稳定匹配中与其结婚的所有男士中她最不喜欢的. 这绝非偶然.

图 12.28　在面向男士支配关系下表 12.4 的优先选择的所有稳定匹配的格

定理 12.12(McVitie 和 Wilson[1971])　*在男士最优稳定匹配中，每一位女士在她的所有稳定婚姻的配偶中有最不喜欢的配偶.*

证明　我们通过矛盾法来证明这一定理. 设 M_m 是男士最优稳定匹配. 假设女士 w 与其在另一个稳定匹配 M 中的配偶 m_2 相比，更喜欢 M_m 中的配偶 m_1. 因为婚姻 $w—m_1$ 发生在 M_m 中，所以与其他稳定匹配，包括 M 中他的任意其他对象相比，m_1 一定更喜欢 w. 注意到这时 m_1 和 w 是 M 中的一个中断对，这导致矛盾. ■

推论 12.12.1　*面向男士格中的最大元素是面向女士格中的最小元素，即男士最优稳定匹配是女士最劣稳定匹配.*

对于面向女士支配关系和格也有类似的结果.

730

最后，让我们看一看，面向男士格中哪个元素是产生于 GS 算法的稳定匹配.

定理 12.13　*在 GS 算法的稳定匹配中，每名男士获得在任意稳定匹配中可以获得的最佳的配偶.*

GS 算法产生面向男士格中的最大(男士最优)元素这一事实没有什么令人惊奇的. 因为男士求婚，而且是从他的第一选择开始向下进行求婚的. 练习 10 给出定理 12.13 的证明.

12.8.4　稳定婚姻的扩展

本节所介绍的稳定婚姻问题有相当的局限性. 即男士和女士的集合大小相同；每一个人的

优先选择列表包含与其性别相反的每一个人；没有平局；婚姻必须是一夫一妻而且是异性的等等. 在本小节以及练习中，我们讨论稳定婚姻问题的若干变形.

【例 12.20 全国居民匹配程序】 当医学院学生从医学院毕业时，他们向医院申请做实习医生. 每名学生(居民)列出医院的等级列表，而医院也列出居民的等级列表. 人们设计开发出全国居民匹配程序(NRMP)，来寻找医院和居民之间的稳定匹配(关于 NRMP 的广泛论述，参见 Roth[1982]).

NRMP 在几个方面不同于稳定婚姻问题. 首先，居民和医院的数量不必相同. 其次，每家医院一般需要匹配多个居民. 注意，这两个条件不是独立的.

我们利用对 GS 算法稍加修改而得的 NRMP 算法寻找上述情况下的稳定匹配. 假设医院 H_i 有 q_i 个居民的定额. 与 GS 算法中的婚约相反，NRMP 算法把等待列表临时提交给一名居民. 当这一算法停止时，等待列表变成实习医生列表.

与 GS 算法相同，NRMP 算法的每一阶段也是一个两步过程，在这一过程中，所有自由(不在等待列表上)的居民申请他们还没有申请过的优先选择列表上的最高等级医院. 然后，每家医院把这些新的申请人与已在它的等待列表中的那些人比较，并在定额以内，基于它的优先选择列表，选择更新的等待列表. 当每名申请人或是在等待列表上或者已遭到优先选择列表上的每家医院的拒绝时，这个算法停止(稳定性的证明类似于 GS 算法产生一个稳定匹配的证明，留作练习).

731

例如，假设有两家医院 H_a 和 H_b，每家医院都在两名居民的定额下接受实习程序，且有 5 名居民 r_1，r_2，r_3，…，r_5. 居民和医院有表 12.7 所示的优先选择列表(注意有一名居民将不被匹配，在男士和女士这样的等大小列表的稳定婚姻问题中，这是不可能的). NRMP 算法的各阶段如表 12.8所示. 因为医院 H_a 与它的两个最高优选居民匹配，这不可能是中断对的一部分. 因此，在这种情况下，NRMP 的结果产生一个稳定匹配.

表 12.7 5 名居民，2 家医院稳定匹配问题的优先选择列表

居民优先选择					医院优先选择	
r_1	r_2	r_3	r_4	r_5	H_a	H_b
H_a	H_b	H_a	H_a	H_a	r_2	r_1
H_b	H_a	H_b	H_b	H_b	r_3	r_3
					r_4	r_5
					r_5	r_2
					r_1	r_4

表 12.8 应用于表 12.7 的优先选择列表的 NRMP 算法

阶段 1		阶段 2		阶段 3		阶段 4	
申请	等待列表：H_a H_b	申请	等待列表：H_a H_b	申请	等待列表：H_a H_b	申请	等待列表：H_a H_b
r_1-H_a	r_3 r_2	r_1-H_b	r_3 r_1	r_1-	r_2 r_1	r_1-	r_2 r_1
r_2-H_b	r_4	r_2	r_4 r_5	r_2 H_a	r_3 r_5	r_2-	r_3 r_5
r_3-H_a		r_3-		r_3-		r_3-	
r_4-H_a		r_4-		r_4-		r_4-H_b	
r_5-H_a		r_5-H_b		r_5-		r_5-	

除了上面所提到的不同之处外，NRMP 算法还包含不是每家医院都在每个人的优先选择列表上的可能性. 稍微做一下修改，就有可能允许每家医院提供仅包含其申请人的优先选择列表. ■

稳定婚姻问题和实习医生匹配问题的共同点是，(稳定)匹配是在两组不同对象之间进行的：男士和女士，居民和医院. 例如，当学校分配室友时，他们仅有一组工作对象(男生或女生). 寻找室友的稳定匹配问题归功于 Gale 和 Shapley[1962].

【例 12.21 稳定室友】 考虑有四个人要分配到两间宿舍，2 人一室的情况. 每个人有其他三个人的严格优先选择等级. 必须做稳定的宿舍分配. 即基于他们的优先选择列表，没有两个人能够离开他们的各自室友而更友好地住在一起. 假设他们的优先选择列表如下所示：

732

$\underline{p_1}$	$\underline{p_2}$	$\underline{p_3}$	$\underline{p_4}$
p_4	p_1	p_2	p_2
p_2	p_3	p_4	p_1
p_3	p_4	p_1	p_3

如果配对是 $R=\{p_1\text{-}p_2, p_3\text{-}p_4\}$，那么 p_1 和 p_4 都愿意离开他们各自的室友而住在一起；在 p_4 的列表中 p_1 高于 p_3，而在 p_1 的列表上，p_4 高于 p_2. 因此，R 不是稳定匹配. 然而不难看出 $S=\{p_1\text{-}p_4, p_2\text{-}p_3\}$是稳定匹配. ■

稳定室友问题是 Donald Knuth 于 1976 年提出的 12 个未解问题中的一个. 参见 Knuth[1997]. 他问到：根据某个优先选择顺序，$2n$ 个人的集合中的每个人可以等级化其他 $2n-1$ 个人. 如果可能的话，寻找一个有效的算法，使得我们可以得到一个 n 对的稳定集合. Irving[1985]给出了 Knuth 问题的答案，他寻找到了一个多项式时间算法，这个算法产生一个稳定匹配，如果这样的匹配存在的话. 与稳定婚姻问题不同，在稳定室友问题中稳定性并非总是可能的. 练习 11 给出一个这样的例子. 关于稳定室友问题的其他参考资料，可以参见 Subramanian[1994]和 Feder[1992].

稳定性问题已进入很多不同的领域. 除了实习医生和室友问题之外，稳定匹配与拍卖、计算机科学(如散列法和数据结构)、数学(如着色和最短路径问题)、息票托收等领域也存在一定的关联(参见 Galvin[1995]、Knuth[1974]、Knuth[1997]、Roth 和 Sotomayor[1990]、Spira[1973]以及 Ullman[1972]).

本节练习

1. 有多少大小为 n 的不同稳定婚姻问题？为什么？
2. 在大小为 n 的稳定婚姻问题中，有多少可能的匹配？请给出解释.
3. 寻找表 12.5 所给出的大小为 4 的稳定婚姻问题的所有稳定匹配.
4. 证明对于表 12.4 所给出的大小为 4 稳定婚姻问题，$M=\{m_1-w_3, m_2-w_2, m_3-w_1, m_4-w_4\}$不是稳定匹配.
5. (a)把 GS 算法运用于表 12.9 所示的稳定婚姻问题.

表 12.9 大小为 4 的稳定婚姻问题

男士的优先选择				女士的优先选择			
$\underline{m_1}$	$\underline{m_2}$	$\underline{m_3}$	$\underline{m_4}$	$\underline{w_1}$	$\underline{w_2}$	$\underline{w_3}$	$\underline{w_4}$
w_1	w_2	w_3	w_4	m_4	m_3	m_2	m_1
w_2	w_1	w_4	w_3	m_3	m_4	m_1	m_2
w_3	w_4	w_1	w_2	m_2	m_1	m_4	m_3
w_4	w_3	w_2	w_1	m_1	m_2	m_3	m_4

(b)把“女士求婚”GS 算法运用于表 12.9 所示的稳定婚姻问题.

6. 制作大小为 3 且正好有一个稳定匹配的稳定婚姻问题.
7. 写出大小为 n 的稳定婚姻问题的匹配的稳定性测试算法.
8. 本练习完成定理 12.11 的证明.

(a)证明(12.14)式中定义的婚姻形成一个匹配.

(b)证明所有匹配都是不同的.

(c)证明(12.14)式中定义的婚姻所产生的匹配是稳定的.

9. 对于任意给定的稳定婚姻问题，证明稳定匹配在面向男士或面向女士支配关系下形成一个格.

10. 本练习通过矛盾法给出定理 12.13 的证明. 设 M_{GS}和 M 是大小为 n 的稳定婚姻问题面向男士格下的两个稳定匹配. 假设男士 m 在 M_{GS}中与 w_i结婚，在 M 中与 w_j结婚，而且假设与 w_i相比 m 更喜欢 w_j.

    (a)解释在这个 GS 算法中，为什么 w_j要解除与 m 的婚约，而与其他人订婚，比如说与 m'订婚.

    (b)在 GS 算法中，不失一般性，假设 w_j' 在某个稳定匹配中选择了 m'而首次破坏了某个稳定婚约. 在所有稳定匹配中的 m'的所有配偶中，证明 m'更喜欢 w_j.

    (c)解释为什么 m'和 w_j 是稳定匹配 M中的中断对(这将是一个矛盾).

11. 考虑表 12.10 的室友问题.

    (a)证明不存在稳定匹配.

    (b)如果人 p_4 改变他的优先选择列表，能够找到稳定的匹配吗?

表 12.10 室友优先选择列表

p_1	p_2	p_3	p_4
p_2	p_3	p_1	p_1
p_3	p_1	p_2	p_2
p_4	p_4	p_4	p_3

733 ~ 734

参考文献

AHUJA, R. K., MAGNANTI, T. L., and ORLIN, J. B., *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, NJ, 1993.

BALINSKI, M., and RATIER, G., "Of Stable Marriages and Graphs, and Strategy and Polytopes," *SIAM Rev., 39* (1997), 575–604.

BERGE, C., "Two Theorems in Graph Theory," *Proc. Natl. Acad. Sci. USA, 43* (1957), 842–844.

BERGE, C., *Graphs and Hypergraphs*, American Elsevier, New York, 1973.

BROGAN, W. L., "Algorithm for Ranked Assignments with Application to Multiobject Tracking," *J. Guidance*, (1989), 357–364.

CHRISTOFIDES, N., *Graph Theory: An Algorithmic Approach*, Academic Press, New York, 1975.

COOK, W. J., CUNNINGHAM, W. H., PULLEYBLANK, W. R., and SCHRIJVER, A., *Combinatorial Optimization*, Wiley, New York, 1998.

DEO, N., *Graph Theory with Applications to Engineering and Computer Science*, Prentice Hall, Englewood Cliffs, NJ, 1974.

DERMAN, C., and KLEIN, M., "A Note on the Optimal Depletion of Inventory," *Management Sci., 5* (1959), 210–214.

DEVINE, M. V., "A Model for Minimizing the Cost of Drilling Dual Completion Oil Wells," *Management Sci., 20* (1973), 532–535.

EDMONDS, J., "Paths, Trees and Flowers," *Canad. J. Math., 17* (1965), 449–467. (a)

EDMONDS, J., "The Chinese Postman Problem," *Oper. Res., 13, S1* (1965), 373. (b)

EDMONDS, J., and JOHNSON, E. L., "Matching, Euler Tours and the Chinese Postman," *Math. Program., 5* (1973), 88–124.

EGERVÁRY, E., "Matrixok Kombinatóricus Tuladjdonságairól," *Mat. Fiz. Lapok, 38* (1931), 16–28. ("On Combinatorial Properties of Matrices," translated by H. W. Kuhn, Office of Naval Research Logistics Project Report, Department of Mathematics, Princeton University, Princeton, NJ, 1953.)

FEDER, T., "A New Fixed Point Approach for Stable Networks and Stable Marriages," *J. Comput. System Sci., 45* (1992), 233–284.

FIALA, J., KRATOCHVÍL, J., and PROSKUROWSKI, A., "Distance Constrained Labeling of Precolored Trees," in A. Restivo, S. Ronchi Della Rocca, and L. Roversi (eds.), *Theoretical Computer Science: Proceedings of the 7th Italian Conference, ICTCS 2001, Torino, Italy, October 2001*, Lecture Notes in Computer Science, 2202, Springer-Verlag, Berlin, 2001, 285–292.

FORD, L. R., and FULKERSON, D. R., *Flows in Networks*, Princeton University Press, Princeton, NJ, 1962.

GALE, D., and SHAPLEY, L. S., "College Admissions and the Stability of Marriage," *Amer. Math. Monthly, 69* (1962), 9–15.

GALLAI, T., "Über Extreme Punkt- und Kantenmengen," *Ann. Univ. Sci. Budap., Eötvös Sect. Math., 2* (1959), 133–138.

GALVIN, F., "The List Chromatic Index of a Bipartite Multigraph," *J. Combin. Theory,*

Ser. B, 63 (1995), 153–158.

GONDRAN, M., and MINOUX, M., *Graphs and Algorithms*, Wiley, New York, 1984.

GRÖTSCHEL, M., LOVÁSZ, L., and SCHRIJVER, A., *Geometric Algorithms and Combinatorial Optimization*, 2nd ed., Springer-Verlag, Berlin, 1993.

GUSFIELD, D., and IRVING, R. W., *The Stable Marriage Problem: Structure and Algorithms*, Foundations of Computing Series, MIT Press, Cambridge, MA, 1989.

735

HALL, P., "On Representatives of Subsets," *J. Lond. Math. Soc., 10* (1935), 26–30.

HOPCROFT, J. E., and KARP, R. M., "A $n^{5/2}$ Algorithm for Maximum Matching in Bipartite Graphs," *SIAM J. Comput., 2* (1973), 225–231.

IRVING, R. W., "An Efficient Algorithm for the Stable Roommates Problem," *J. Alg., 6* (1985), 577–595.

IRVING, R. W., and LEATHER, P., "The Complexity of Counting Stable Marriages," *SIAM J. Comput., 15* (1986), 655–667.

KNUTH, D. E., "Structured Programming with **go to** Statements," *Comput. Surveys, 6* (1974), 261–301.

KNUTH, D. E., *Stable Marriage and Its Relation to Other Combinatorial Problems*, CRM Proceedings & Lecture Notes, 10, American Mathematical Society, Providence, RI, 1997. (Translated by M. Goldstein with revisions by D. Knuth from *Mariages Stables et Leurs Relations Avec d'Autres Problèmes Combinatoires*, Collection de la Chaire Aisenstadt, Les Presses de l'Université de Montréal, Montreal, Canada, 1976.)

KOLITZ, S., personal communication to R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, 1991.

KÖNIG, D., "Graphen und Matrizen," *Mat. Fiz. Lapok, 38* (1931), 116–119.

KUHN, H. W., "The Hungarian Method for the Assignment Problem," *Naval Res. Logist. Quart., 2* (1955), 83–97.

LAWLER, E. L., *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.

LOVÁSZ, L., and PLUMMER, M. D., *Matching Theory*, North-Holland, Amsterdam, 1986.

MASON, A. J., and PHILPOTT, A. B., "Pairing Stereo Speakers Using Matching Algorithms," *Asia-Pacific J. Oper. Res., 5* (1988), 101–116.

MCVITIE, D. G., and WILSON, L. B., "The Stable Marriage Problem," *Comm. ACM, 14* (1971), 486–490.

MICALI, S., and VAZIRANI, V. V., "An $O(\sqrt{|V|} \cdot |E|)$ Algorithm for Finding Maximum Matching in General Graphs," *Proceedings of the Twenty-First Annual Symposium on the Foundations of Computer Science*, IEEE, Long Beach, CA, 1980, 17–27.

MINIEKA, E., *Optimization Algorithms for Networks and Graphs*, Dekker, New York, 1978.

NORMAN, R. Z., and RABIN, M. O., "An Algorithm for a Minimum Cover of a Graph," *Proc. Amer. Math. Soc., 10* (1959), 315–319.

PAPADIMITRIOU, C. H., and STEIGLITZ, K., *Combinatorial Optimization: Algorithms and Complexity*, Prentice Hall, Englewood Cliffs, NJ, 1982.

PETERSON, J., "Die Theorie der Regulären Graphen," *Acta Math., 15* (1891), 193–220.

ROTH, A. E., "The Economics of Matching: Stability and Incentives," *Math. Oper. Res., 7* (1982), 617–628.

ROTH, A. E., and SOTOMAYOR, M. A. O., *Two-Sided Matching*, Cambridge University Press, Cambridge, MA, 1990.

SPIRA, P. M., "A New Algorithm for Finding All Shortest Paths in a Graph of Positive Arcs in Average Time $O(n^2 \log^2 n)$," *SIAM J. Comput., 2* (1973), 28–32.

SUBRAMANIAN, A., "A New Approach to Stable Matching Problems," *SIAM J. Comput., 23* (1994), 671–700.

TUCKER, A. C., *Applied Combinatorics*, Wiley, New York, 1980.

TUTTE, W. T., "The Factorization of Linear Graphs," *J. Lond. Math. Soc., 22* (1947), 107–111.

736

ULLMAN, J. D., "A Note on the Efficiency of Hashing Functions," *J. Assoc. Comput. Mach., 19* (1972), 569–575.

第 13 章　图和网络的优化问题

本章给出解决若干个组合优化问题的过程. 在本书中，我们已经遇到过这样的问题，而且在 12.4 节～12.6 节中对其中的一些做了详细的讨论. 我们把讨论过的问题翻译成涉及图或有向图的问题. 更一般地，我们把它们翻译成图或有向图的问题，对于这些图或有向图中的每一条边或弧可以指定一个或多个非负实数. 这样的图和有向图分别称为网络和有向网络.

还应该提到的是，我们选择已知有好解(好算法)的问题来讨论. 并非所有组合优化问题都有好解. 2.4 节和 11.5 节所讨论的卖货郎问题就是一个这样的例子. 对于这样的情况，正如我们在 2.18 节指出的那样，我们寻找在特殊情况下能够解决这一问题的好算法，或者给出近似解. 而关于这些方法的讨论则超出了本书的范围.

13.1　最小支撑树

13.1.1　Kruskal 算法

在例 3.28～例 3.32 中，我们描述了寻找带权图，即网络 G 的最小支撑树的问题. 最小支撑树是形成一棵树的 G 的子图，它具有如下性质：没有其他支撑树在其边上有更小的权和. 本节讨论寻找网络 G 的最小支撑树的一个简单算法. 这一算法或者寻找一棵最小支撑树，或者得出该网络没有支撑树的结论. 根据定理 3.19，网络没有支撑树意味着这个网格是非连通的. 寻找最小支撑树的问题有非常广泛的应用：例如，在大规模运输、通信、分布式网络等的筹划中，在简化数据存储中，以及在数据挖掘等方面都有应用. 我们在 3.5.4 节给出其中的一些应用. 关于更多应用的概览和参考资料，参见 Ahuja，Magnanti 和 Orlin[1993]、Graham 和 Hell[1985].

737

根据 Graham 和 Hell[1985]，最早的寻找最小支撑树的算法要追溯到 Borůvka[1926a，b]，⊖ Boruvka 对电线网络的最经济的布局感兴趣. 这一问题的更早期工作归功于人类学家 Czekanowski[1909，1911，1928]，他在关于分类表的工作中对此作了研究.

让我们回想并继续处理例 3.29 的问题. 假设在一个偏远的地方有 5 个村落，且它们之间的道路距离如图 13.1 所示. 我们希望用下面的方法沿着现存的道路铺设电话线：每一对村落通过电话服务器被连接起来，而且我们要使电话线的总长度最小. 这个问题相当于寻找图 13.1 中的网络的最小支撑树的问题.

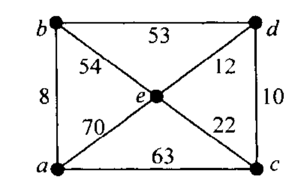


图 13.1　村落之间的道路距离网

在寻找有最小化边的权和这样的网络的最小支撑树的过程中，可以一次构建这棵树的一条边，这个过程将是贪婪的(greedy)；即首先加入最小权的边. 可以证明贪婪是行得通的. 特别地，以权递增的顺序排列网络 G 的边. 在这个例子中，我们如下排列它们：$\{a, b\}$，$\{c, d\}$，$\{d, e\}$，$\{c, e\}$，$\{b, d\}$，$\{b, e\}$，$\{a, c\}$，$\{a, e\}$. 在权相等时，顺序是随意的. 我们以这一顺序检查这些边，确定每一条边是否应该被包含在这一支撑树中. 只要它不与已包含的边形成回路，它就被包含于这一支撑树中. 这一算法称为 Kruskal 算法(这一算法归功于 Kruskal[1956])，算法概括如下.

⊖ Nešetřil，Milková 和 Nešetřilová[2001]是用捷克语写成的 Borůvka 的关于最小支撑树的著作的英文翻译，其中还有评论和历史发展等.

算法 13.1：Kruskal 的最小支撑树算法

输入： $n>1$ 个顶点的网络 G.

输出： 定义 G 的最小支撑树的边的集合，或 G 非连通的信息.

步骤 1. 按权的递增顺序排列 G 的边(在权相等时顺序是随意的.)集合 $T=\varnothing$. (T 是最小支撑树的边的集合).

738 **步骤 2.** 如果 G 的每一条边已检查过，停止，并输出 G 是非连通的信息. 否则，检查排序列表中第一条未检查的边，并将其放入 T 中当且仅当它不与已在 T 中的边形成回路. 如果这条边被加入 T 中，进入步骤 3. 否则重复步骤 2.

步骤 3. 如果 T 有 $n-1$ 条边，停止，并输出 T. 否则，进入步骤 2.

定理 13.1 *如果 G 是 n 个顶点的连通网络，Kruskal 算法将终止于一个有 $n-1$ 条边的最小支撑树 T. 如果 G 是非连通网络，那么算法在检查所有边之后，T 中仍没有 $n-1$ 条边，这时它将停止并输出 G 是非连通的信息.*

在给出定理 13.1 的证明之前，我们利用图 13.1 的例子来说明这一算法. 我们从把边$\{a, b\}$放入 T 中开始. 这时，G 的顶点的数量 $n=5$，T 还没有 $n-1=4$ 条边，所以我们返回步骤 2. 在步骤 2，我们注意到并不是 G 的每一条边都已检查过. 我们检查边$\{c, d\}$，把它加入到 T 中，因为它不与 T 中的边形成回路. 现在，T 有两条边，所以我们返回到步骤 2. 因为第一个未检查边$\{d, e\}$不与 T 中现存的边形成回路，我们把这条边加入到 T 中. 现在 T 中有三条边，所以我们返回到步骤 2. 下一条被检查的边$\{c, e\}$与 T 的边$\{c, d\}$和$\{d, e\}$形成一条回路，所以我们不把它加入到 T 中. 我们对边$\{b, d\}$重复步骤 2. 这条边被加入到 T 中. 现在，因为 T 有四条边，我们停止，并得出这些边定义 G 的一棵最小支撑树的结论.

注意，我们可以把 Kruskal 算法运用到没有权的图 G，并以任意顺序简单地列出它的所有边. 这时，这一算法将产生 G 的支撑树，或得出 G 非连通的结论.

我们来考虑 Kruskal 算法的计算复杂度. 忽略把边按顺序排列的步骤. 这个算法至多需要 e 步，每一条边需要一步. 读者应该注意，不要把这里的"步"与算法中的"步"混淆起来. 有些边需要算法中的步骤 2 和 3. 所以，在某种意义下，总共需要 $2e$ 步. 然而，$O(e)=O(2e)$(参见 2.18.1 节)，所以我们认为 e 步和 $2e$ 步是一样的.

按权增加的顺序排列边需要多少步？这是 3.6.4 节所讨论的排序问题. 在那节，我们观察到，排序有 e 项的集合可用 $ke\log 2e$ 步完成，其中 k 是一个常数. 因此，整个 Kruskal 算法至多用 $e+ke\log 2e\leqslant e+ke^2$ 步完成. 注意，$e+ke^2\leqslant n^2+kn^4$，因为 $e\leqslant\left(\frac{n}{2}\right)\leqslant n^2$. 因此，完成整个 Kruskal 算法的步数是 e 或 n 的多项式(使用 2.18 节的术语，粗略讨论表明我们有一个 $O(e\log_2 e)$ 的算法. Tarjan[1984a]给出的 Kruskal 的最好实现是用 $O(e\alpha(n, e))$ 步骤寻找最小支撑树，其中，实际中函数 $\alpha(n, e)$ 增加得非常缓慢，可以认为是一个小于 6 的常数. 参见 Ahuja，Magnanti 和 Orlin[1993]、Cheriton 和 Tarjan[1976]、Graham 和 Hell[1985]、Tarjan[1983，1984b].)

【例 13.1 聚类和数据挖掘(Ahuja，Magnanti 和 Orlin[1993]、Gower 和 Ross[1969]、Zahn [1971])】 739 在诸如检测、决策或模式识别，特别是今天很多领域中巨大数据表的挖掘等很多实际问题中，我们通常设法把数据分成自然组，或簇(cluster). 聚类问题也出现在下列各应用领域：分子数据库、天体物理学研究、地理学信息系统、软件度量、全球生态监控、医药以及电讯诈骗分析和信息卡数据诈骗分析等(参见 Arratia 和 Lander[1990]、Baritchi，Cook 和 Holder[2000]、

Church[1997]、Godehart[1990]、Karger[1997]、Neil[1992]). 其思想是同簇中的元素的关系应该亲密，而不同簇中的元素应该没有如此亲密的关系，然而如何精确地定义这一关系取决于使其精确的各种设想(例如，参见 Guénoche，Hansen 和 Jaumard[1991]、Hansen，Frank 和 Jaumard[1989]、Hansen 和 Jaumard[1987]、Hansen，Jaumard 和 Mladenovic[1998]、Mirkin[1996]). 在很多应用中，被分类对象被表示成二维欧几里得空间(更一般地，k 维空间)中的点. 在这一空间中，Kruskal 算法被广泛用于聚类. 在每一次迭代中，算法产生一个边的集合，这个集合把图划分成树的集合. 可以把每一棵树考虑为一个簇. 如果存在 n 个顶点，这个算法产生 n 个划分，这一算法从每一个簇有一个顶点的划分开始，并结束于由所有顶点组成的唯一一个簇的划分. 这些划分中哪个是“最好的”或“最有用的”依赖于“对象函数”的精确定义. ■

13.1.2 定理 13.1 的证明[⊖]

为了证明定理 13.1，注意，在这个算法的每一步，T 中的边定义一个没有回路的图. 假设这一算法终止于 T 有 $n-1$ 条边. 那么，根据定理 3.20，T 是一棵树，从而，是 G 的一棵支撑树. 因此，G 是连通的. 所以，如果 G 是非连通的，那么这个算法将终止于没有寻找到有 $n-1$ 条边的集合 T.

如果 G 是连通的，我们证明这个算法给出一棵最小支撑树. 首先，我们证明这个算法的确给出支撑树. 然后我们再证明这个支撑树是最小的. 如果这个算法给出有 $n-1$ 条边的 T，那么根据定理 3.16，我们第一个观察结果自然成立. 假设这个算法终止时没有给出有 $n-1$ 条边的 T. 在这一算法停止时，我们考虑 T 中的边. 这些边定义 G 的一个支撑子图，我们也将其称为 T. 这一子图是非连通的，因为 G 的任意连通支撑子图必须至少有 $n-1$ 条边，其中有一棵支撑树. 因此，子图 T 至少有两个分支，但是因为 G 是连通的，所以在 G 中存在一条连接 T 的两个不同分支中的顶点的边 $\{x, y\}$. 现在 $\{x, y\}$ 不与 T 中的边形成回路. 因此，当这一算法开始检查这条边时，它应该包含于 T. 因此，只有当这个算法不能得以正确运用时，才能出现这种情况. 740

现在，我们知道这个算法给出一棵支撑树 T. 设 S 是 G 的一棵最小支撑树. 如果 $S=T$，我们就已完成证明. 假设 $S\neq T$. 因为 $S\neq T$，且因为二者有相同数量的 $n-1$ 条边，那么在 T 中一定存在一条边不在 S 中. 以权递增排序边，寻找第一条在 T 中但不在 S 中的边 $e_1=\{x, y\}$.

因为 S 是 G 的支撑树，所以在 S 中存在一条从 x 到 y 的简单链 $C(x, y)$. 现在，把边 e_1 加入到 S 中给出一个回路. 因此，因为 T 没有回路，所以在这条回路中一定存在一条不在 T 中的链 $C(x, y)$ 上的边 e_2. 设 S' 是从 S 通过移去 e_2 增加 e_1 而得的边的集合. 这时，S' 定义一个连通图(为什么)，而且有 n 个顶点和 $n-1$ 条边，所以根据定理 3.16，S' 是支撑树. 我们考虑两种情况.

情况 1：在以权递增的边的排序中，边 e_2 在边 e_1 的前面. 对于这种情况，e_2 不能放入到 T 中，所以它一定与 T 中在 e_2 之前已被检查的边形成回路 D. 但是，e_1 是 T 的不在 S 中的第一条边，所以 D 中的所有边一定在 S 中，因为它们在 T 中. 因此，e_2 不能放入在 S 中，这是一个矛盾. 情况 1 不可能.

情况 2：边 e_1 超前于边 e_2. 在这种情况下，e_2 的权至少与 e_1 的权一样大. 因此，S' 的权总和至多等于 S 的权总和，因为 S 是最小支撑树，所以，S' 是最小支撑树. 另外 S' 与 T 的公共边比 S 与 T 的公共边多 1 条.

如果 $T\neq S'$，我们重复对 T 和 S' 的讨论，得到一棵最小支撑树 S''，这棵最小支撑树与 T 的公共边比 S' 与 T 的公共边多一条. 最后，我们找到与 T 相同的最小支撑树. 完成证明.

⊖ 本小节可以跳过.

13.1.3 Prim算法

还有另一个寻找最小支撑树的算法，这一算法也是贪婪算法. 在此，我们给出这一算法.

算法 13.2：Prim的最小支撑树算法㊀

输入： $n>1$ 个顶点的网络 G.

输出： G 的一棵最小支撑树，或 G 非连通的信息.

步骤 1. 设置集合 $T=\varnothing$. 选出 G 的任意一个顶点并将其放入到树 T 中.

步骤 2. 把连接 T 中的顶点与不在 T 中的顶点的所有边中权最小的边加入到 T 中. 如果权最小的边有多条，则任选其一. 如果不可能把任意一条边加入到 T 中，那么停止并输出 G 是非连通的信息.

步骤 3. 如果 T 有 $n-1$ 条边，停止并输出 T. 否则重复步骤 2.

为了说明这一算法，我们再一次考虑图 13.1 中的图，并从顶点 a 开始. 于是，我们把边$\{a, b\}$加入到 T，因为它是连接顶点 a 的边中权最小的边. 接下来，我们检查连接顶点 a 或 b 与不在 T 中的顶点 c，d 或 e 的边. 在这些边中，边$\{b, d\}$的权最小；我们把它加入到 T 中. 接下来，我们检查连接顶点 a，b 或 d 与不在 T 中的顶点 c 或 e 的边. 在这些边中，$\{d, c\}$的权最小. 我们把这条边加入到 T 中. 最后，我们检查连接 a，b，c 或 d 和剩下的不在 T 中的顶点 e 的边. 在这些边中，边$\{d, e\}$的权最小. 我们把它加入到 T 中. 现在，T 有四条边，我们停止. 注意，通过使用 Kruskal 算法，我们发现了相同的 T.

741

定理 13.2 *如果 G 是 n 个顶点的连通网络，那么 Prim 算法停止于有 $n-1$ 条边的最小支撑树 T. 如果 G 是非连通网络，那么这一算法停止并输出 G 是非连通的信息，因为不可能向 T 加入另一条边.*

这一定理的证明留作练习(练习 14).

Prim 算法的最好实现是执行 $O(e+n\log_2 n)$步(参见 Ahuja，Magnanti 和 Orlin[1993]、Graham 和 Hell[1985]、Johnson[1975]、Kershenbaum 和 Van Slyke[1972]). 关于 Prim 算法的另一个版本，请参见 Gabow 等[1986]，这个算法是目前寻找最小支撑树的最快算法. 另一个早前的寻找最小支撑树的未发表算法是由 Sollin 发现的(参见 Ahuja，Magnanti 和 Orlin[1993]). Yao[1975]对 Sollin 的算法做了改进，开发了一个 $O(e\log\log n)$的最小支撑树的算法. 练习 16 给出另一个最小支撑树的算法，它源于 Borůvka[1926a，b]的工作，且适于所有的权都不同的情况.

本节练习

1. 对于图 13.2 中的每一个网络，利用 Kruskal 算法寻找最小支撑树.

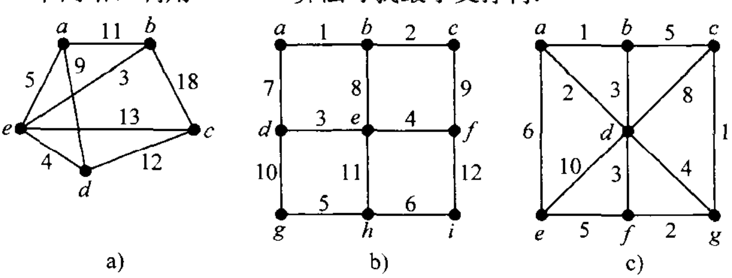


图 13.2 13.1 节练习的网络

㊀ 这一算法是由 Prim[1957]发现的，通常也把这一发现归功于他. 事实上，似乎早前这一算法已被 Jarnik[1930]发现(Graham 和 Hell[1985]).

2. 利用 Prim 算法，从顶点 a 开始，重复练习 1.
3. 对于图 13.3 中的每一个网络运用 Kruskal 算法.
4. 对于图 13.3 中的每一个网络，从顶点 a 开始，运用 Prim 算法.

742

5. 一家化工公司有八个存储容器并希望开发一种管道系统，使得可以在任意容器之间流动化学材料. 以英尺计算，每对容器之间的距离如表 13.1 所示. 确定在哪些容器对之间构建管道，使得化学材料可以在任意容器之间流动，且所使用的管道总长度最小.

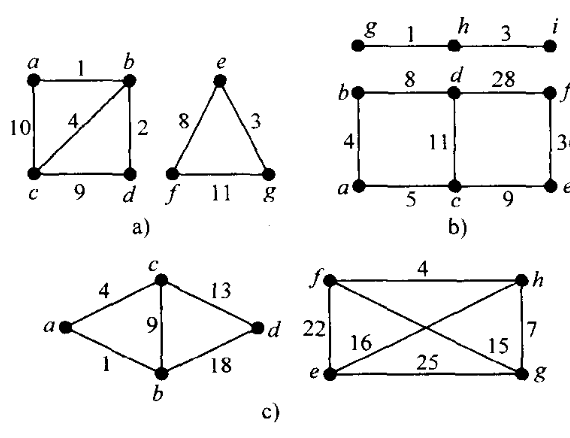


图 13.3　13.1 节练习的网络

表 13.1　容器间的距离



容器	1	2	3	4	5	6	7	8
1	0	3.9	6.3	2.7	2.1	5.4	6.0	4.5
2	3.9	0	2.7	5.4	3.6	7.8	6.9	3.3
3	6.3	2.7	0	7.8	5.1	7.5	5.7	3.0
4	2.7	5.4	7.8	0	2.1	4.8	4.5	2.7
5	2.1	3.6	5.1	2.1	0	2.7	3.3	2.4
6	5.4	7.8	7.5	4.8	2.7	0	1.8	3.0
7	6.0	6.9	5.7	4.5	3.3	1.8	0	1.5
8	4.5	3.3	3.0	2.7	2.4	3.0	1.5	0

6. 我们使用高频电路(电线)(参见例 3.30)连接数字计算机的各种组件. 以毫米为单位的每一对组件之间的距离如表 13.2 所示. 确定要连通哪些组件对，使得组件集合是连通的，且组件之间的电线的总长度最小(以降低电容量和延迟效应).
7. 对于图 13.4 中的点集合，使用例 13.1 的方法，基于这些点之间的实际距离寻找分成 8 个簇的，7 个簇的，…，1 个簇的所有划分.

743

表 13.2　组件之间的距离

	1	2	3	4	5	6
1	0	6.7	5.2	2.8	5.6	3.6
2	6.7	0	5.7	7.3	5.1	3.2
3	5.2	5.7	0	3.4	8.5	4.0
4	2.8	7.3	3.4	0	8.0	4.4
5	5.6	5.1	8.5	8.0	0	4.6
6	3.6	3.2	4.0	4.4	4.6	0

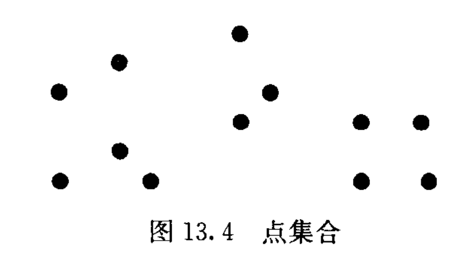


图 13.4　点集合

8. 使用城市之间的直线距离，寻找连接华盛顿、DC 和每一个(陆地)美国州府的最小支撑树(在因特网上可以查找到这些城市之间的距离).
9. 对于图 13.2 中的每一个网络，寻找一棵最大支撑树(maximum spanning tree)，所谓最大支撑树就是不比任意其他支撑树的权和小的支撑树.
10. 修改 Kruskal 算法使得它可以寻找最大支撑树.
11. 如果网络的每条边都有不同的权，能否有多棵最小支撑树？为什么？
12. (a)对于图 13.1 中的网络，寻找包含边$\{a, e\}$的最小支撑树(当要求我们包含一条特殊的电话线或已经存在一条线时，会出现这样的问题).

    (b)对于边$\{c, e\}$重复(a).
13. 如何修改 Kruskal 算法使其用于处理诸如练习 12 中的情况，其中一条或多条边已被指定为必须属于这个支撑树？

14. 证明 Prim 算法可行.
15. 粗略说明 Prim 算法的计算复杂度由 e 或 n 的多项式定界.
16. 在 n 顶点的网络 G 中寻找最小支撑树 T 的另一种算法如下所示，这里要求所有权都不相同.

   步骤 1. 设置集合 $T=\varnothing$.

   步骤 2. 设 G' 是由 T 中的边组成的 G 的支撑子图.

   步骤 3. 对于 G' 的每一个连通分支 K，寻找 G 中连接 K 的顶点与 G' 的某个其他分支的顶点的最小权边. 如果不存在这样的边，停止并输出 G 是非连通的信息. 否则把所有这些新边加入到 T 中.

744

   步骤 4. 如果 T 有 $n-1$ 条边，停止并输出 T. 否则重复步骤 2.

   这一算法源自 Borůvka[1926a，b].

   (a)使用这一算法寻找图 13.2 中的每一个网络的最小支撑树.

   (b)在所有权不同的情况下，证明这一算法可行.
17. 证明：最小支撑树 T 是唯一的，当且仅当，当把不在 T 中的任意边$\{x, y\}$加入到 T 中形成一个回路时，$\{x, y\}$的权大于该回路上所有其他边的权.
18. 设 G 是连通图. G 中的简单割集(simple cut set)是满足下面条件的边的集合 B：这些边的删除使得 G 非连通，而且使得 B 的任意真子集都没有这一性质. 设 F 是 G 的边的集合. 证明下面的事实.

   (a)如果 F 是简单割集，F 满足下面的性质 C：对于 G 的每一棵支撑树 H，F 的某条边在 H 中.

   (b)如果 F 满足性质 C 但 F 的任意真子集不满足性质 C，那么 F 是简单割集.
19. (Ahuja，Magnanti 和 Orlin[1993])设 w_{ij} 是边$\{i, j\}$上的权，如何寻找最小化下面式子的最小支撑树 T?
$$\left[\sum_{\{i,j\}\in T}(w_{ij})^2\right]^{1/2},$$
20. (Ahuja，Magnanti 和 Orlin[1993])两棵支撑树 T 和 T' 是*邻接的*(adjacent)，如果它们除一条边之外所有边都相同. 证明：对于任意两棵支撑树 T' 和 T''，我们可以寻找下面的支撑树的序列：
$$T'=T_1,T_2,T_3,\cdots,T_k=T'',$$
   其中 T_i 邻接于 T_{i+1}，$i=1, 2, \cdots, k-1$.
21. (Ahuja，Magnanti 和 Orlin[1993])假设在网络 G 中每一条边都被着红色或蓝色.

   (a)说明如何寻找有最多红色边数的支撑树.

   (b)假设某棵支撑树有 k' 条红色边，另一棵支撑树有 $k''>k'$ 条红色边. 证明：对于满足 $k''>k>k'$ 的 k，存在有 k 条红色边的支撑树.

13.2 最短路径问题

13.2.1 问题

本节考虑寻找(有向)网络中两个顶点之间的最短路径问题. 我们从能够说明这一问题的若干例子开始.

745

【例 13.2 州际公路】 假设你希望仅利用州际公路开车从纽约到洛杉矶. 这时，选取的最短路径是什么呢? 可以把这一问题翻译成如下的网络问题. 设图的顶点是州际公路的汇合点，当存在直接连接两个汇合点的州际公路时，用一条边连接两个这样的顶点. 把一个实数放到边$\{x, y\}$上，表示顶点 x 和 y 之间经过州际公路的英里数. 在最终的网络中，设链的*长度*(length)被定义为它的边上的数值(权)的总和，而顶点 x 和 y 间的*距离*(distance)$d(x, y)$是 x 和 y 之间的最短链的长度(如果 x 和 y 之间不存在链，则距离无定义). 我们寻找纽约和洛杉矶之间的链，它有最小长度，即它的长度等于 d(纽约，洛杉矶). ■

【例 13.3 计划空中旅行】 假设你想从纽约飞往曼谷. 使你在空中停留时间最少的路径是什么呢? 为了回答这一问题，设有向图的顶点是全球空运网络中的城市，当存在直接从 x 到 y 的直飞航班时从 x 到 y 画一条弧. 在弧(x, y)上放置表示飞行时间的实数. 然后，定义这一有向网络中从 x 到 y 的路径的长度为这些弧上的数值(权)总和，定义从 x 到 y 的距离 $d(x, y)$是从 x 到 y 的最短路径的长度(如果不存在从 x 到 y 的路径，则距离无定义). 我们寻找从纽约到曼谷的最短长度的路径，即长度等于 d(纽约，曼谷)的路径. ■

一般地，我们研究网络或有向网络，寻找从顶点 x 到顶点 y 的最短链或路径. 我们集中精力讨论有向网络. 在 13.2.2 节，我们将给出在有向网络中寻找从 x 到 y 的最短路径的算法. 首先，我们来说明基本想法. 在图 13.5 的有向网络中，路径 x，a，b，y 有长度 $1+5+1=7$. 路径 x，c，d，b，y 更短；它的长度等于 $3+1+1+1=6$. 实际上，这是从 x 到 y 的最短路径. 因此，$d(x, y)=6$. 注意，我们说 x，c，d，b，y 是从 x 到 y 的一条最短路径. 从 x 到 y 的长度等于$d(x, y)$的路径可能不只一条. 这里，x，e，d，b，y 是另一条这样的路径. 还需注意，$d(y, x)$无定义；不存在从 y 到 x 的路径.

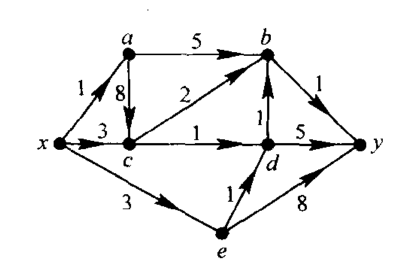


图 13.5　从 x 到 y 的最短路径是 x，c，d，b，y

746

在(有向)网络中寻找从顶点 x 到顶点 y 的最短路径问题是常见的组合问题. 根据 Goldman [1981]，这可能是在政府部门最常遇到的组合问题. 具 Goldman 评估：仅由美国交通署都市大众运输办事处一家政府机构开发的最短路径算法一年要使用十亿次.

令人惊讶的是，很多问题都可以转述为最短路径问题. 我们给出几个例子.

【例 13.4 TEX 文档处理系统(Ahuja，Magnanti 和 Orlin[1993])】 数学科学中广泛使用的文档处理系统 TEX 把文章的段落分解成左右对齐、形成"最优"外观的行. (注意，段落的最后一行只需左对齐.)为了弄明白如何做到这一点，设段落按顺序有字 1，2，…，n. 设 $c_{i,j}$表示从字 i 开始到字 $j-1$ 结束的行的"无吸引性"($c_{i,j}\geqslant 0$). 例如，$c_{i,j}$可以评估段落的格式化宽度与字 i，$i+1$，…，$j-1$ 的总长度之间的绝对差. 这一问题是把段落分解成行使得段落的总"成本"最小. 为了弄明白这为何是一个最短路径问题，设有向网络的顶点是字加上标记为 $n+1$ 的一个"虚设"顶点(字). 包括从 i 到所有 $j>i$ 的弧. 弧(i, j)的成本是 $c_{i,j}$. $1=i_1$，i_2，…，$i_k=n+1$ 是从字 1 到字 $n+1$ 的一条最短路径，这意味着对于 $p=1$，2，…，$k-1$，使用开始于字 i_p，而结束于字 $i_{p+1}-1$ 的行，将会得到最吸引人的外观样式. (注意，顶点 $n+1$ 必须包含在其中，使得无论怎么定义每一行的结束都能够包含字 n.)[⊖] ■

【例 13.5 结构钢梁的库存(Ahuja，Magnanti 和 Orlin[1993]、Frank[1965])】 一家建筑公司需要各种不同长度的结构钢梁. 为了节省存储空间和维持较大库存的成本，这家公司把较长钢梁切割成较短钢梁，而不是存储所有所需长短的钢梁. 库存的钢梁长度依赖于对不同长短钢梁的需求. 另外，切割操作将浪费一些钢材. 这家公司如何决定在库存中保存什么样长度的钢梁以便最小化建立库存的总成本，及切割所浪费的可用钢材总成本呢? 我们可以把这一问题阐述成最短路径问题，其方法是：对应于可能需要的不同长度钢梁，使用顶点 0，1，…，n 定义一个有向网络. 我们假设长度为 L_i 的钢梁比长度为 L_{i+1}的钢梁短，而长度为 L_0 的钢梁长度为 0. 从顶点 i 到每一个顶点 $j>i$ 我们包括一条弧. 我们把弧(i, j)解释成这样的策略：在库存中保存长度

⊖ 我们忽略了允许带连字符号的字处于一行末端的复杂情况. 在练习 27 中，我们要求读者描述这一复杂情况.

747 为 L_j 的钢梁，并使用它们来满足对长度为 L_{i+1}，L_{i+2}，…，L_j 钢梁的需求. 从顶点 0 到顶点 n 的路径对应于在库存中保存的钢梁长度的集合. 假设 D_i 代表对长度为 L_i 的钢梁的需求. 弧(i, j)的成本 $C_{i,j}$ 由下式给出：

$$C_{i,j} = K_j + C_j \sum_{k=i+1}^{j} D_k, \tag{13.1}$$

其中 K_j 是建立保存长度为 L_j 的钢梁的库存设备的成本，C_j 是长度为 L_j 的钢梁的成本，上式中的第二项对应于用长度为 L_j 的钢梁来满足长度从 L_{i+1} 到 L_j 的所有钢梁需求的成本. 如果我们想要最小化总成本，那么从顶点 0 到顶点 n 的最短路径给出在库存中保存的钢梁分类. 注意，这一方法考虑较长的钢梁成本(可能会)较高，以及切割它们时造成的浪费. 这样的分析简化了什么呢? [见练习 20.] ■

有很多其他最短路径的例子. 我们将在 13.2.3 节给出若干其他的例子. Ahuja，Magnanti 和 Orlin[1993]对此有更多的讨论和引用.

13.2.2 Dijkstra 算法

我们给出的最短路径算法来自于 Dijkstra[1959]. 下面给出有向网络 D 的这一算法. 设 $w(u, v)$是弧(u, v)的权. 回想一下，我们的标准假设是，网络中的弧的权是非负的. 我们将需要这种假设. 在不存在从 u 到 v 的弧时，设置 $w(u, v)$为∞很方便. 算法的基本思想是，在第 k 次迭代寻找 k 个顶点 u_1，u_2，…，u_k，这些顶点是此网络中 k 个最靠近 x 的顶点，即有 k 个最短距离 $d(x, u_1)$，$d(x, u_2)$，…，$d(x, u_k)$的顶点. 对于每一个 u_j，我们还要寻找从 x 到 u_j 的最短路径. 对于 k 解决了这一问题之后，我们对于 $k+1$ 解决这一问题.

从第 k 次迭代到第 $k+1$ 次迭代的一般思想是，对于不在 k 个最近顶点中的每一个顶点 v，对于所有 j，我们计算 $d(x, u_j)+w(u_j, v)$，并寻找第$(k+1)$个最近顶点作为使该和为最小的 v. 这样做的基本原理是，如果 x，a_1，a_2，…，a_p，v 是从 x 到 v 的最短路径，那么 x，a_1，a_2，…，a_p 必定是从 x 到 a_p 的最短路径. 如果不是，那么我们可以使用从 x 到 a_p 的更短路径来寻找从 x 到 v 的更短路径. 另外，如果权都是正的，如果 v 是到 x 的第$(k+1)$个最近顶点，且 x，a_1，a_2，…，a_p，v 是从 x 到 v 的最短路径，那么 a_p 一定在 k 个最近顶点当中. 即使权是 0，也可以选择第$(k+1)$个顶点 v，使得存在一条从 x 到 v 的最短路径 x，a_1，a_2，…，a_p，v，且 a_p 是 k 个最近顶点中的一个.

在 Dijkstra 算法的每一步，我们保存包含于前 k 次迭代中的顶点列表，这定义一个类 W，我们还保存用于从 x 到 u_j 的最短路径中的弧的列表，这定义一个类 B，而且对于所有的 j，我们保存 $d(x, u_j)$的记录. 一旦 y 被加入到 W 中，我们停止. 然后，我们使用 B 来构建这一路径. 现

748 在我们已经准备好给出这一算法的更加形式的描述.

算法 13.3：Dijkstra 最短路径算法㊀

输入：有向网络 D 及 D 中的顶点 x 和 y.

输出：从 x 到 y 的最短路径，或不存在从 x 到 y 的路径的信息.

步骤 1. 初始化，把顶点 x 放入到类 W 中，设置 $B=\varnothing$，设置 $d(x, x)=0$.

步骤 2.

步骤 2.1. 对于 W 中每一个顶点 u 和不在 W 中的每一个顶点 v，设置 $\alpha(u, v)=d(x, u)+w(u, v)$. 寻找 W 中的 u 和不在 W 中的 v，使得 $\alpha(u, v)$最小(在相等的情况下可以随意选择).

步骤 2.2. 如果步骤 2.1 中 $\alpha(u, v)$的最小值为∞，那么停止，并给出不存在从 x 到 y 的路

㊀ 在此，我们给出这一算法的基本想法. 以后，我们描述如何改进它.

径的信息.

步骤2.3. 如果步骤2.1中$\alpha(u, v)$的最小值不是∞，那么把v放入W(v是下一个选择的顶点)，把弧(u, v)放入B，并设置$d(x, v)=\alpha(u, v)$.

步骤3. 如果y也不在W中，返回到步骤2. 如果y在W中，停止. 通过使用从x到y的B的唯一弧路径，可以得到从x到y的最短路径. 其方法是从y开始向回寻找这条最短路径.

我们使用图13.6中的有向网络来说明这一算法. 这一算法中的每个步骤如表13.3所示[表中没有给出值为无穷的$\alpha(u, v)$]. 例如，在迭代2中，弧(x, a)是计算的值中最小的值，所以把a加入到W中，而且把(x, a)加入到B中，取$d(x, a)$等于$\alpha(x, a)$；在迭代4中，$\alpha(c, e)$是最小值，所以把e加入到W中，且把(c, e)加入到B中，取$d(x, e)$等于$\alpha(c, e)$；以此类推. 在第7步迭代中，顶点y被加入到W中. 利用B中的弧，我们从y开始向后运算. 我们发现从d可以到达y，从a可以到达d，从x可以到达a. 因此，我们发现路径x, (x, a), a, (a, d), d, (d, y), y是从x到y的最短路径.

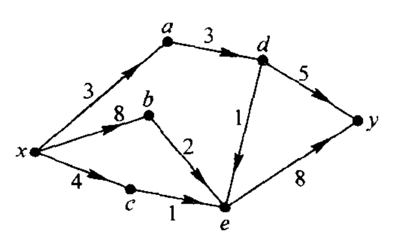


图13.6 有向网络

表13.3 运用于图13.6中的有向网络的Dijkstra算法

迭代	计算的有限数$\alpha(u, v)$	加入W	加入B	新$d(x, v)$
1	无	x	—	$d(x, x)=0$
2	$\alpha(x, a)=0+3=3$ $\alpha(x, b)=0+8=8$ $\alpha(x, c)=0+4=4$	a	(x, a)	$d(x, a)=3$
3	$\alpha(x, b)=0+8=8$ $\alpha(x, c)=0+4=4$ $\alpha(a, d)=3+3=6$	c	(x, c)	$d(x, c)=4$
4	$\alpha(x, b)=0+8=8$ $\alpha(a, d)=3+3=6$ $\alpha(c, e)=4+1=5$	e	(c, e)	$d(x, e)=5$
5	$\alpha(x, b)=0+8=8$ $\alpha(a, d)=3+3=6$ $\alpha(e, y)=5+8=13$	d	(a, d)	$d(x, d)=6$
6	$\alpha(x, b)=0+8=8$ $\alpha(e, y)=5+8=13$ $\alpha(d, y)=6+5=11$	b	(x, b)	$d(x, b)=8$
7	$\alpha(e, y)=5+8=13$ $\alpha(d, y)=6+5=11$	y	(d, y)	$d(x, y)=11$

现在我们再评述一下这一算法的计算复杂度. 这一算法至多需要n次迭代，其中n是这个有向网络的顶点数量. 对于每一次迭代都加入一个顶点. 每一次迭代包含加法$d(x, u)+w(u, v)$，其中对在W中的顶点u和不在W中的顶点v的每一对顶点需要一次加法，即总共至多需要n^2次加法. 同样，在至多n^2个数$d(x, u)+w(u, v)$中寻找最小数也可以通过至多n^2次比较完成. 因此，对每一次迭代，这一算法至多需要$2n^2$步. 总之，这一算法至多需要$2n^3$步，它有一个多项式界. 使用2.18节的术语，我们有一个$O(n^3)$算法.

实际上，通过在这个过程中做几处简单的改动，这个算法就改善为一个$O(n^2)$算法[⊖]. 假设

⊖ 本节的其余部分可以跳过.

我们令 u_k 是第 k 个最近的顶点. 我们取 $u_1=x$. 对于所有 $v\neq x$, 令 $\alpha_1(v)=\infty$, $\alpha_1(x)=0$, 且定义

$$\alpha_{k+1}(v)=\min\{\alpha_k(v),\alpha_k(u_k)+w(u_k,v)\}. \tag{13.2}$$

于是不难证明(练习 25),

749~750

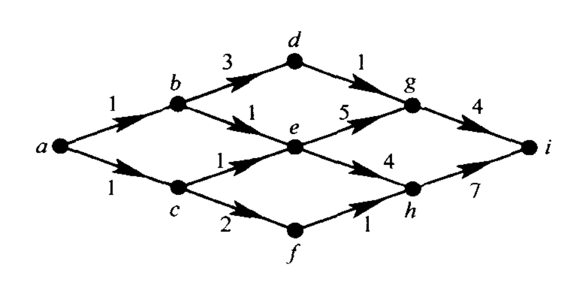
$$\alpha_{k+1}(v)=\min\{d(x,u_j)+w(u_j,v):j=1,2,\cdots,k\}. \tag{13.3}$$

在每一次迭代时, 我们计算 n 个数 $\alpha_{k+1}(v)$. 对于每一个 v, 我们通过首先做一个加法, 然后寻找两个数的最小值来计算这 n 个数. 然后, 我们寻找数 $\alpha_{k+1}(v)$ 的集合中的最小值, 这一过程至多需要 n 步. 给出最小值的顶点 v 是 u_{k+1}(在相等的情况下, 任选一个). 现在, 加入 u_{k+1} 的迭代时的总步数至多是 $n+2$, 所以整个算法所需要的总步数至多等于 n^2+2n. 如果我们只使用标签 $\alpha_k(v)$, 我们必须更加小心地计算集合 B. 特别地, 在 $\alpha_{k+1}(v)$ 减小的每一步, 我们将必须记录顶点 u_j, 使得 $\alpha_{k+1}(v)$ 被重新定义为 $d(x, u_j)+w(u_j, v)$. 此时, 当 u_{k+1} 选取为 v 时, 对应的 (u_j, v) 被加入到 B 中. 对 Dijkstra 算法的改进版的详细描述留给读者. 在不同假设下对 Dijkstra 算法的各种实现和改进的讨论, 参见 Ahuja, Magnanti 和 Orlin[1993].

13.2.3 对调度问题的应用

尽管我们使用距离的术语阐述了最短路径问题, 但是权不一定表示距离. 最短路径算法的许多应用适合于弧对应于某种活动, 而弧上的权对应于活动成本的情况. 这类问题包括寻找开始于某个出发点的活动序列, 实现一个期望的目标, 最小化总成本. 另一方面, 权也可以是实施活动所需的时间, 且这一问题是寻找在最小的时间总量内实现期望目标的活动序列. 我们在 11.6.3 节遇到过类似的问题. 对于这些情况, 网络有时被称为 PERT(Project Evaluation and Review Tehnique)网或 CPM(Critical Path Method)网. 我们利用下面的例子说明这些想法.

【例 13.6 制造过程】 制造过程开始于一块原木材. 这一木材必须被切割成某种形状, 剥皮, 打孔并上漆. 切割必须在打孔之前进行, 而剥皮必须在上漆之前进行. 假设切割需要花费 1 个时间单位; 剥皮需要 1 个时间单位; 对未切割木材上漆需要 2 个时间单位, 而对已切割木材上漆需要 4 个时间单位; 未剥皮的木材打孔需要 3 个时间单位, 而对已剥皮但未上漆的木材打孔需要 5 个时间单位, 对已上漆木材打孔则需要 7 个时间单位. 我们的问题是寻找在尽可能短的时间内完成这一过程的活动序列. 我们可以让有向网络 D 的顶点表示制造过程的阶段, 例如, 原木材; 已切割木材; 切割和打孔; 切割, 剥皮和打孔等等. 如果一个单一活动占用的过程从阶段 i 到阶段 j, 那么我们从阶段 i 到阶段 j 画一条弧. 然后, 我们在弧 (i, j) 上放置权, 对应于从 i 到 j 所需的时间总量. 所讨论的有向网络如图 13.7 所示. 例如, 存在从 b 到 d 和 e 的弧, 因为切割木材之后可以是打孔或剥皮. 弧 (b, d) 的权等于 3, 因为它对应于在未剥皮的木材上打孔. 我们在这个有向网络中寻找从原木材顶点 (a) 到已剥皮、切割、打孔和上漆顶点 (i) 的最短路径. ■



关键字:

a. 原木材, *b*. 切割, *c*. 剥皮, *d*. 切割和打孔, *e*. 切割和剥皮, *f*. 剥皮和上漆, *g*. 切割、打孔和剥皮, *h*. 切割、剥皮和上漆, *i*. 剥皮、切割、打孔和上漆

图 13.7 一个制造过程的有向网络. 顶点对应于这一过程中的各阶段, 弧对应于一个阶段到另一个阶段所需的活动, 权对应于这些活动所需的时间

751

【例 13.7 生产线上的次品检查(Ahuja, Magnanti 和 Orlin[1993]、White[1969])】 一条生产

线有 n 个阶段，且在同一时间内有一批产品流经这条生产线. 在每一个阶段，都可能出现次品. 在每一个阶段进行检查的成本很高，但是只在阶段 n 结束时检查，则意味者大量的生产时间可能花费在继续生产已经有缺陷的产品上. 检查次品的优化计划是什么呢？假设在未覆盖所有次品的意义下检查是"完美的"，而且次品不可修理并立即被丢弃. 假设 B 是传送到这条生产线上的产品批次的大小，a_i 是在阶段 i 生产次品的概率，p_i 是在阶段 i 生产一个产品的成本，$f_{i,j}$ 是在阶段 i 做上一次检查后在阶段 j 后检查这批产品的固定成本，而 $g_{i,j}$ 则是在阶段 i 做上一次检查后在阶段 j 后检查一个产品的成本.（注意，检查成本依赖于什么时候做的上一次检查，因为次品可以在所有中间阶段出现.）我们可以如下计算在阶段 i 结束时非次品的期望数量 B_i：

$$B_i = B\prod_{k=1}^{i}(1-a_k).$$

为了把我们的问题阐述为最短路径问题，设有向网络的顶点是阶段 1，2，…，n，及对应于开始的额外顶点 0，并对于每一个 i 和每一个 $j>i$，有一条从阶段 i 到阶段 j 的弧. 一条路径对应于检查它的顶点(除顶点 0 之外)的计划. 我们所关心的弧(i, j)的成本由下式给出：

752

$$c_{i,j} = f_{i,j} + B_i g_{i,j} + B_i \sum_{k=i+1}^{j} p_k. \tag{13.4}$$

这表示当我们在阶段 i 检查，其后在阶段 j 检查时，在阶段 $i+1$，$i+2$，…，j 的总成本. 上式的前两项是在阶段 j 后检查的固定成本和可变成本，第三项是阶段 $i+1$ 到阶段 j 的生产成本. 从顶点 0 到顶点 n 的最短路径给出最廉价的检查调度. ■

本节练习

1. 证明：在有向图中，从 x 到 y 的最短路径一定是简单路径.
2. 证明：在图中，从 x 到 y 的最短链一定是简单链.
3. 在图 13.8 中的每一个有向网络中，使用 Dijkstra 算法(如算法 13.3 的描述)寻找从 a 到 z 的最短路径.
4. 寻找例 13.6 的问题中的最有效的制造过程.
5. 一种产品必须经过打磨、抛光、称重和检查. 打磨必须在抛光和称重之前，而抛光必须在检查之前. 打磨需要 7 个时间单位，抛光需要 10 个时间单位，对未抛光产品称重需要 1 个时间单位，而对已抛光产品称重需要 3 个时间单位，对未称重产品检查需要 2 个时间单位，而对已称重产品检查需要 3 个时间单位. 最快的生产调度是什么？
6. 一家公司要投资一个车队并尝试着决定保留汽车时间的最佳方案. 5 年后，它将卖出所有剩余汽车并让一家外围公司提供运输. 在策划下一个 5 年计划时，这家公司评估在年 i 的开始买进汽车并在年 j 的开始卖出汽车，将有净成本 a_{ij}（购入价减去折旧加上运营和维修成本）. 以千美元计数的值 a_{ij} 由下面的矩阵给出：

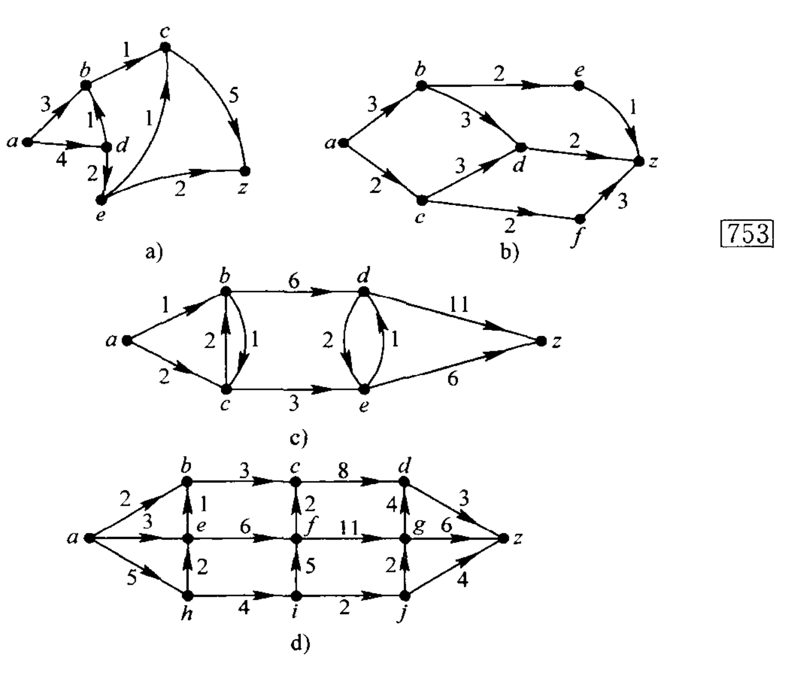


图 13.8 13.2 节和 13.3 节练习的有向网络

753

$$(a_{ij}) = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccccc} 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 4 & 6 & 9 & 12 & 20 \\ & 5 & 7 & 11 & 16 \\ & & 6 & 8 & 13 \\ & & & 8 & 11 \\ & & & & 10 \end{bmatrix} \end{array}.$$

为了决定什么时候买进和卖出汽车的最便宜的策略，我们设有向网络的顶点是数 1，2，3，4，5，6，且对于 $i<j$ 包含所有弧(i, j)，并设权 $w(i, j)$是 a_{ij}. 弧(i, j)可解释为在年 i 的开始买进一辆汽车并在年 j 的开始卖出它. 寻找最便宜的策略.

7. 在例 13.5 中，假设有 8 种钢梁长度可用. 每一长度对应于有向网络 N 中的一个顶点. 下面的图表包含需求(D_j)，库存设备成本(K_j)，以及每一种长度的钢梁的成本(C_j)：

j	0	1	2	3	4	5	6	7
L_j	0	2	4	8	16	32	64	128
D_j	0	4	6	16	2	8	5	4
K_j	0	1	2	3	4	5	6	7
C_j	0	10	19	27	34	40	45	49

N 中的大部分弧的成本 $C_{i,j}$ 由下面的矩阵给出：

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array}\begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} & 4 & 10 & 26 & X & 36 & 41 & 45 \\ & & 6 & 22 & 24 & 32 & 37 & 41 \\ & & & 16 & 18 & Y & 31 & 35 \\ & & & & 2 & 10 & 15 & Z \\ & & & & & 8 & 13 & 17 \\ & & & & & & 5 & 9 \\ & & & & & & & 4 \\ & & & & & & & \end{bmatrix} \end{array}.$$

(a)使用(13.1)式计算这个矩阵中缺失的弧成本(X，Y，Z).

754

(b)寻找这个有向网络中从顶点 0 到顶点 7 的最短路径. 哪些钢梁应该保存在库存中？

8. 图 13.9 给出一个通信网络[弧(i, j)对应于一个链接，通过这个链接 i 与 j 可以直接通信]. 假设弧(i, j)的权 p_{ij} 是从 i 到 j 的链接有效的概率. 假设链接中出现的故障彼此互相独立，在一条路径上的所有链接有效的概率是各链接概率的积. 寻找从 a 到 z 的最可靠的路径[提示：考虑 $-\log p_{ij}$].

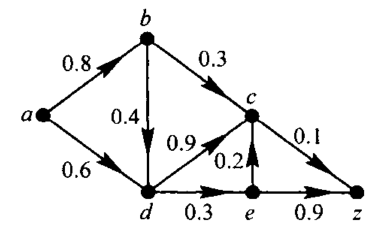


图 13.9 通信网络

9. 假设你计划在 20 年后退休，并想要为退休投资 10 万美元，而且你知道在 i 年的可用投资选项是 $A(i, 1)$,$A(i, 2)$，…，$A(i, k_i)$，且投资 $A(i, j)$可得回报率 $r(i, j)$并在 $y(i, j)$年后到期(当然，这经过了简化). 将这一寻找最优投资策略问题阐述为最短路径问题.

10. 在最短路径的搜索中，考虑从 i 到 j 的所有可能路径不是高效的. 为了说明这一点，假设 D 有 n 个顶点并从每一个顶点到每一个其他顶点有弧. 如果 x 和 y 是 D 的任意两个顶点，寻找从 x 到 y 的路径数量.

11. 如果每一条弧的成本增加 u 个单位，从 a 到 z 的最短路径的成本会增加 u 的倍数吗？为什么？

12. 在例 13.4 中，假设我们想要创建有 6 个字的一个段落. 对于每对 i 和 j，$i<j\leqslant 7$，下面的矩阵给出成本 $c_{i,j}$(“字”7 对应于虚设顶点).

	1	2	3	4	5	6	7
1		8	4	0	5	10	15
2			7	3	2	7	12
3				7	3	3	8
4					10	6	9
5						2	4
6							0
7							

(a)构建网络 N，它的顶点是字(包括"字"7)，对于 $j>i$ 它的弧是 (i, j)，每一条弧 (i, j) 有权 $c_{i,j}$.

(b)求 N 中从字 1 到字 7 的最短路径.

(c)描述如何将 6-字段落格式化.

755

13. 在例 13.7 中，假设在生产线上有 6 个阶段. 使用下面的固定、可变和生产成本等式，在阶段 i 检查及下一次在阶段 j 检查的总成本由等式(13.4)给出：

$$f_{i,j} = (j-i)$$

$$g_{i,j} = (j-i)\frac{i}{j}$$

$$(p_k) = (5,8,4,7,2,6)$$

$$B_i = 0.8(1.1)^{-i}.$$

寻找最廉价的生产调度.

14. (Bondy 和 Murty[1976])一只狼、一只山羊和一棵卷心菜同处于一个河岸上，船夫要把它们运往对岸，但是一次只能运其中的一个. 狼和山羊不能同时留在这条河的岸边，山羊和卷心菜也不能. 船夫如何用最少的时间总量把它们运到对岸？

15. (Bondy 和 Murty[1976])一名男士有一个装满了 8 加伦葡萄酒的酒壶和分别可以盛装 5 加伦和 3 加伦酒的空壶. 这个男士怎样以最少的步骤把酒分成等量的两份？

16. (Ahuja，Magnanti 和 Orlin[1993])你拥有一笔夏季租赁资产，而且在夏季一开始，你就收到很多使用这笔资产的请求. 每一个请求给出租赁的开始和结束日期. 你的 x 天租赁利率由函数 $f(x)$ 给出. 你如何决定接受哪些请求，以便最大化这个夏季的收入？把这一问题阐述为最短路径问题.

17. (Ahuja，Magnanti 和 Orlin[1993]、Ravindran[1971])图书馆希望最小化特定藏书的置架成本. 在这一藏书中，一定数量的书籍的高度为 H_i，$i=1, 2, \cdots, n$，且 $H_i<H_{i+1}$. 我们想把相同高度的书放在一起，而且以高度递增的顺序放置书籍. 假设各种书的厚度已知. 于是我们可以确定放置高度为 H_i 的书籍所需的书架长度 L_i. 订制高度为 H_i 的书架的成本 F_i 是固定的，它与订制的书架长度无关，高度为 H_i 的书架每英寸长度的成本是 C_i. 因此，如果我们订制总长度为 x_i 的高度为 H_i 的书架，那么这种书架的成本是$F_i+C_ix_i$. 注意，因为每一种订制有固定的成本，所以我们也许不订制每一个高度的书架，因为对于 $j>i$，我们总可以把高度为 H_i 的书籍放置在高度为 H_j 的书架上. 我们应该如何订制书架？把这一问题阐述为最短路径问题.

18. 回想计划空中旅行的问题(例 13.3). 除了花在飞行上的时间之外，考虑存在与每一个城市相关的"成本"的情况，该成本表示在那一城市的平均滞留时间.

(a)仅使用弧权阐述这一新"问题"为标准的最短路径问题.

(b)对于图 13.5 中的有向网络解决这一问题. 其中，每一个城市的成本(平均滞留时间)由下表给出：

城市	a	b	c	d	e
成本	1	2	5	4	3

756

19. 如果一个网络的所有弧有不同的权，这个网络一定有唯一的最短路径吗？给出证明或反例.

20. (Ahuja，Magnanti 和 Orlin[1993])我们对例 13.5 中的结构钢梁库存问题的阐述在某些地方做了简化.

(a)假设要把长度为 5 的钢梁切割成长度为 2 的钢梁，会发生什么问题？

(b)在这个例子中有哪些其他简化？

(c)如果我们能够把单一钢梁切割成若干等长的钢梁(例如，把一个长度为 14 的钢梁切割成三个长度为 4 的钢梁和一些浪费的钢材)，我们应该如何修改这一分析？

(d)如果我们不仅能把一个钢梁切割成若干等长的钢梁，而且还能出售边角料而得到剩余价值，其价格是每单位长度 d 美元，那么我们如何修改这一分析？

21. 设 D 是有向图．我们定义距离$\overline{d}(x, y)$为通过在有向图 D 的每一条弧上放置权 1 而得到的有向网络中从 x 到 y 的最短路径的长度．讨论如何利用 D 的邻接矩阵的幂计算$\overline{d}(x, y)$.
22. 一个网络中从 a 到 z 的最短路径不一定是唯一的，从 a 到 z 的路径中它也可能没有弧的最小数量．然而在所有从 a 到 z 的最短路径中，Dijkstra 算法能否产生弧的数量最小的最短路径？为什么？
23. 在 Dijkstra 算法中，证明：如果忽略弧的方向，那么集合 B 的弧定义一棵以顶点 x 为根的树．
24. 描述一个在有向网络 D 中寻找从给定顶点 x 到其他每一个顶点的最短路径的算法．
25. 证明(13.2)式定义的 α_{k+1} 满足(13.3)式．
26. (a)使用标签 $\alpha_k(v)$写出 Dijkstra 算法的 $O(n^2)$版本的详细描述．

    (b)把这一算法运用于图 13.8 中的每一个有向网络，寻找从 a 到 z 的最短路径．
27. 在例 13.4 和练习 12 中，我们没有考虑用连字符连接一个字，以便更好地优化段落的外观．你如何修改例 13.4 的文档处理系统的有向网络的阐述以适应连字符？

13.3　网络流

13.3.1　最大流问题

假设 D 是有向网络，并设 $c_{ij}=w(i, j)$是弧(i, j)的非负权．本节，我们称 c_{ij} 为弧(i, j)的容量(capacity)，并把它解释成在稳定状态下每单位时间内可以流经弧的某种物品的最大量．物品可能是已完成的产品、信息、人、油、卡车、信件、电流等等．

757

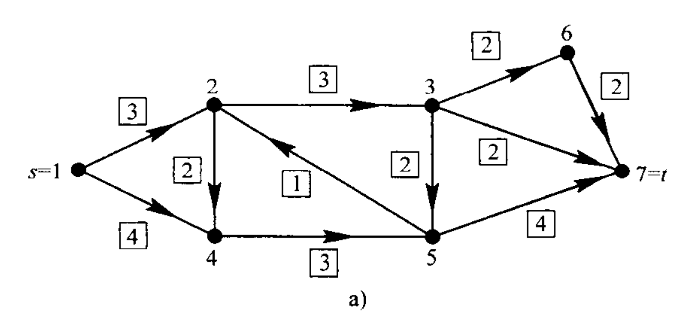
流只允许沿着弧的方向流动，即从 i 到 j．我们固定一个源头(source)顶点 s 和一个汇聚(sink)顶点 t，并认为一个流开始于 s 结束于 t(在我们的所有例子中，s 都没有进入弧，而 t 没有出去的弧．但不必假设这些性质)．设 x_{ij} 是经过弧(i, j)的流．这时，我们要求

$$0 \leqslant x_{ij} \leqslant c_{ij}. \qquad (13.5)$$

这就是说流是非负的，且不能超过容量．我们还有一个守恒定律(conservation law)，这个守恒定律指出对于所有的顶点 $i\neq s, t$，进入到 i 的物质必须从 i 出去，即

$$\sum_j x_{ij} = \sum_j x_{ji}, \quad i \neq s,t. \qquad (13.6)$$

满足(13.5)式和(13.6)式的数集 $x=\{x_{ij}\}$称为一个(s, t)可行流((s, t)-feasible flow)，或简称为(s, t)流，或只称流(flow)．例如，考虑图 13.10 中的有向网络．在这个图的 a 中，容量 c_{ij} 是每一条弧上的方格中所示的数．在 b 中，一个流如圆圈内的数字所示．注意，(13.5)式和(13.6)式成立．例如，$c_{24}=2, x_{24}=0$，所以有 $0\leqslant x_{24}\leqslant c_{24}$．还有，$\sum_j x_{3j} = x_{35} + x_{36} + x_{37} = 2 + 1 + 0 = 3$，



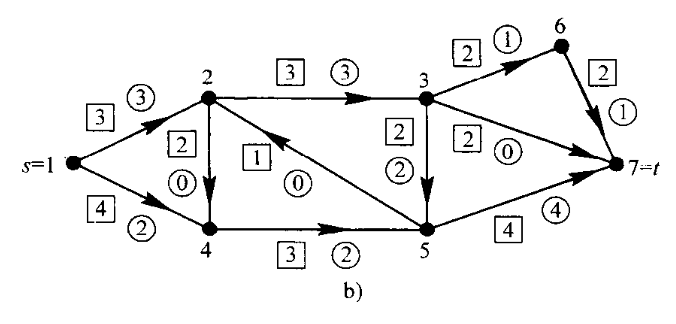


图 13.10　方格内的数值表示容量，圆圈内的数值表示流

$\sum_j x_{j3} = x_{23} = 3$ 等等.

假设 x 定义一个流. 设

$$v_t = \sum_j x_{jt} - \sum_j x_{tj}$$

及

$$v_s = \sum_j x_{js} - \sum_j x_{sj}.$$

注意，如果我们对于所有 i，对下面的项求和：

$$\sum_j x_{ji} - \sum_j x_{ij},$$

那么，等式(13.6)告诉我们所有这些项都是 0，只有 $i=s$ 和 $i=t$ 时除外. 因此，

$$\sum_i \left[\sum_j x_{ji} - \sum_j x_{ij} \right] = v_t + v_s. \tag{13.7}$$

然而，(13.7)式的左边与下式相等：

$$\alpha = \sum_{i,j} x_{ji} - \sum_{i,j} x_{ij}. \tag{13.8}$$

因为(13.8)式中的两个和是对所有弧的所有流求和，所以它们相等，因此 $\alpha=0$. 我们得出 $v_s = -v_t$ 的结论. 因此，存在一个数 v，使得

$$\sum_j x_{ji} - \sum_j x_{ij} = \begin{cases} -v & \text{如果 } i = s \\ v & \text{如果 } i = t \\ 0 & \text{如果 } i \neq s,t. \end{cases} \tag{13.9}$$

在我们图 13.5b 的例子中，这个数 v 等于 5. 它被称为**流值**(value). 如果使用这个流，那么这个流值表示，在给定的时间周期内在这个网络中可以传送的物品的总量. 我们要讨论的问题是：寻找有最大值的流，即**最大流**(maximum flow).

网络流理论的经典参考书是 Ford 和 Fulkerson[1962]所写的书. 其他全面的参考资料是下面的文献：Ahuja，Magnanti 和 Orlin[1993]、Berge 和 Ghouila-Houri[1965]、Cook 等[1998]、Frank 和 Frisch[1971]、Hu[1969]、Iri[1969]、Lawler[1976]、Minieka[1978]、Papadimitriou 和 Steiglitz[1982].

尽管我们给出的例子都是针对有向网络的，但本节给出的所有结果都适用于无向网络. 用两条弧(i, j)和(j, i)替换有容量 c_{ij} 的每条(无向的)边$\{i, j\}$，并设其中的每一条弧都有与边$\{i, j\}$相同的容量.

758 ~ 759

大量的组合优化问题可以阐述为网络流问题. 下面的例子就说明这一点. 对于网络流的许多实际应用，读者可以参考诸如下面的书籍及综述文章，如 Ahuja，Magnanti 和 Orlin[1993]、Aronson[1989]、Bazaraa，Jarvis 和 Sherali[1990]、Glover 和 Klingman[1997]、Gondran 和 Minoux[1984].

【例 13.8 大学招生】 一所大学的招生官有一份申请人列表. 每名申请人都提供一份他能够考虑的可能的专业列表. 招生官的工作是接纳第一轮申请人，使得申请人所考虑的每一个专业正好有一个人允许入学，并使得来自州 i 的允许入学的学生不超过 q_i. 这能够做到吗？构建一个有向网络，其顶点表示申请人、专业和州，再加上一个源头顶点 s 和汇聚顶点 t. 对于每一个专业 m 有弧(s, m)，只要申请人 a 对专业 m 感兴趣就有弧(m, a)，如果申请人 a 居住在州 i 则有弧(a, i)，对于每一个州 i 有弧(i, t). 设弧(i, t)有容量 q_i，而其他所有弧有容量 1. 在这个有向网络中，我们寻找从 s 到 t 的最大流. 如果这个最大流值等于申请人感兴趣的专业数量，那么就有可能解决这一问题. 如果不是，则不能解决这一问题. 其证明留给读者(练习 31). ■

13.3.2 分割

设 S 和 T 是划分有向图 D 的顶点集合的两个集合，即 $V(D)=S\cup T$，$S\cap T=\varnothing$. 我们称划分(S, T)为一个分割(cut). 等价地，我们把分割考虑为从 S 的顶点到 T 的顶点的所有弧的集合 C. 这个集合 C 被称为一个分割，因为除去 C 的弧之后，从 S 的任意顶点到 T 的任意顶点都不存在路径. 如果 x 是 S 的任意顶点，y 是 T 的任意顶点，那么 C 称为一个(x, y)分割((x, y)-cut). 例如，在图 13.10 中，$S=\{1, 2\}$，$T=\{3, 4, 5, 6, 7\}$是一个分割，而且这个分割等价于弧的集合$\{(1, 4), (2, 3), (2, 4)\}$[我们没有把弧$(5, 2)$包括进来，因为它是从 T 的顶点到 S 的顶点的弧]. 注意，如果顶点的数量大于 2，那么总是至少存在两个(x, y)分割：$S=\{x\}$，$T=V(D)-\{x\}$，以及 $S=V(D)-\{y\}$，$T=\{y\}$.

在一个有向网络中，如果(S, T)是一个分割，我们可以定义它的容量(capacity)为$c(S, T)=\sum_{i\in S}\sum_{j\in T}c_{ij}$. 在我们上面的例子中，$c(S, T)=c_{14}+c_{23}+c_{24}=9$. 注意，在我们的例子中，流的值是 5，而这个分割的容量是 9，这个数更大. 下面的结果表明这不是偶然的.

定理 13.3　在一个有向网络中，任意的(s, t)流的值≤任意的(s, t)分割的容量.

证明　设 x 是(s, t)流，(S, T)是一个(s, t)分割. 注意，如果 $i\in S$，且 $i\neq s$，则 $\sum_j x_{ij}-\sum_j x_{ji}$ 等于 0，而且如果 $i=s$ 则其结果是 v. 因此，

760

$$
\begin{aligned}
v &= \sum_j x_{sj}-\sum_j x_{js}\\
&= \sum_{i\in S}\Big[\sum_j x_{ij}-\sum_j x_{ji}\Big]\\
&= \sum_{i\in S}\sum_{j\in S}[x_{ij}-x_{ji}]+\sum_{i\in S}\sum_{j\in T}[x_{ij}-x_{ji}]\\
&= \sum_{i\in S}\sum_{j\in S}x_{ij}-\sum_{i\in S}\sum_{j\in S}x_{ji}+\sum_{i\in S}\sum_{j\in T}[x_{ij}-x_{ji}] \qquad (13.10)\\
&= \sum_{i\in S}\sum_{j\in T}(x_{ij}-x_{ji}), \qquad (13.11)
\end{aligned}
$$

这是因为(13.10)式中的前两项相同. 因此，根据(13.11)式，任意流的值是经过任意分割的净流. 因为 $x_{ij}\leqslant c_{ij}$，$x_{ij}\geqslant 0$，我们有

$$x_{ij}-x_{ji}\leqslant x_{ij}\leqslant c_{ij},$$

所以(13.11)式表明

$$v\leqslant \sum_{i\in S}\sum_{j\in T}c_{ij}=c(S,T).$$

■

推论 13.3.1　在一个有向网络中，如果(S, T)是一个(s, t)分割，而且 x 是一个(s, t)流，那么

$$v=\sum_{i\in S}\sum_{j\in T}[x_{ij}-x_{ji}].$$

证明　这是上述证明的推论. ■

图 13.11 给出图 13.10a 中的网络的另一个(s, t)流. 这个流有值 6. 注意，如果 $S=\{1, 2, 4\}$且 $T=\{3, 5, 6, 7\}$，那么 $c(S, T)=c_{23}+c_{45}=6$. 现在，不可能有值大于这个分割的容量 6 的(s, t)流. 因此，所示的流是一个最大流. 类似地，这个分割一定是最小容量的(s, t)分割，即最小分割(minimum cut). 实际上，当我们发现一

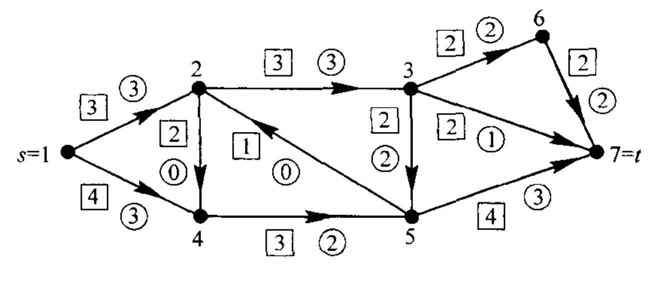


图 13.11　图 13.10a 中的有向网络的另一个流

个流的值与一个分割的容量相同的流和分割时，同样的事情一定成立.

定理 13.4 如果 x 是一个(s,t)流，其值为 v，而(S,T)是一个(s,t)分割，其容量为 $c(S,T)$，如果 $v=c(S,T)$，那么 x 是最大流，而(S,T)是最小分割.

我们已在定理 12.4 中使用了类似的推理，在那里，我们指出因为一个图的任意匹配中边的数量小于或等于任意覆盖中顶点的数量，所以如果我们找到大小相同的匹配和覆盖，那么该匹配一定是最大的，而该覆盖一定是最小的.

对于流的情况，我们在本节强调了有向网络中的分割. 然而，我们可以考虑无向网络 N 中的分割. 与有向网络的情况一样，N 中的分割是把顶点划分成两个集合 S 和 T，或等价地，从 S 的顶点到 T 的顶点的所有弧的集合. 如果 $s\in S$ 且 $t\in T$，那么(S,T)称为(s,t)分割. [761]

如下面的例子所示，寻找网络中最小分割本身也有意义.

【例 13.9 双处理器计算机上的分布式计算(Ahuja，Magnanti 和 Orlin[1993]、Stone[1977])】 最简单的分布式计算系统有两个处理器，我们希望以最小化处理器间通信和计算的成本方式把程序的不同模块(子程序)指定给这些处理器. 在处理器 1 上执行模块 i 的成本是 a_i，在处理器 2 上的执行模块 i 的成本是 b_i. 如果模块 i 和 j 被指定给不同的处理器且这些模块相互作用，那么就存在一个相对较高的处理器间通信成本 c_{ij}. 我们设法给两个处理器分配程序模块，使得最小化总成本，而总成本是处理成本和处理器间通信成本的和. 构建一个无向网格 N，其中源头 s 表示处理器 1 而汇聚 t 表示处理器 2，这个程序的每一个模块有一个顶点. 对每一模块 i 加入边$\{s,i\}$，它有容量 b_i，对每一个模块 i 加入边$\{i,t\}$，其容量为 a_i. 最后，如果在程序执行中模块 i 和 j 相互作用，那么加入一个有容量 $c_{i,j}$ 的边$\{i,j\}$. 设 $E(N)$是 N 中边的集合. 我们注意到处理器的模块分配与(s,t)分割之间存在一一对应，而且一个分割的容量与对应的分配的成本相同. 为了弄明白这一点，假设 A_1 是由分配给处理器 1 的所有模块组成的，A_2 是由分配给处理器 2 的所有模块组成的. 这时，这一分配的成本是

$$\sum_{i\in A_1} a_i + \sum_{i\in A_2} b_i + \sum_{\substack{\{i,j\}\in E(N)\\ i\in A_1, j\in A_2}} c_{i,j}.$$

对应于这一分配的(s,t)分割是由集合$\{s\}\cup A_1$，$\{t\}\cup A_2$ 组成的. 因此，对于 $i\in A_1$，这一分割包含容量为 a_i 的边$\{i,t\}$，对于 $i\in A_2$，它包含容量为 b_i 的边$\{s,i\}$，而对于相互作用模块，A_1 中的 i 和 A_2 中的 j，它有容量为 $c_{i,j}$的边$\{i,j\}$. 这正好是分配的成本. 因此，我们所寻求的最小成本分配对应于网络 N 中的最小(s,t)分割(我们注意到这个例子过于简单化了实际的问题. 事实上，处理成本和处理器间通信成本是用不同单位度量的，实践中的一个主要挑战是“换算”这些成本，使得它们可以进行比较). ⊖ ■ [762]

13.3.3 一个不完善的最大流算法

我们的目标是描述一个寻找最大流的算法. 首先，我们给出一个非常直观的方法，尽管不太完美. 如果 P 是 s 到 t 的简单路径，我们称它为(s,t)路径((s,t)-path)，而且我们设对应的单位流(unit flow)x^P 由下式给出：

$$x_{ij}^P=\begin{cases}1 & \text{如果弧}(i,j)\text{在 } P \text{ 中}\\ 0 & \text{否则}\end{cases}$$

这一算法的思路是持续地添加单位流.

⊖ 对于这一观察结果，我们要感谢 David Roberts.

我们说一个弧(i, j)经过一个流x是未饱和的(unsaturated)如果$x_{ij} < c_{ij}$，且使用$s_{ij} = c_{ij} - x_{ij}$来定义裕量(slack). 我们的基本思路是，如果θ是P的弧中的最小裕量，那么我们能够把单位流x^P增加θ倍，得到一个值增加了θ的新流. 下面就是这一算法.

算法 13.4：最大流算法——第一次尝试

输入： 有源头s和汇聚t的有向网络.

输出： 假想的最大(s, t)流x.

步骤 1. 对于所有的i，j，设置$x_{ij} = 0$.

步骤 2.

步骤 2.1. 寻找一个所有弧都未饱和的(s, t)路径P. 如果不存在这样的路径，进入步骤 3.

步骤 2.2. 计算P的每一个条弧的裕量.

步骤 2.3. 计算P的弧的最小裕量θ.

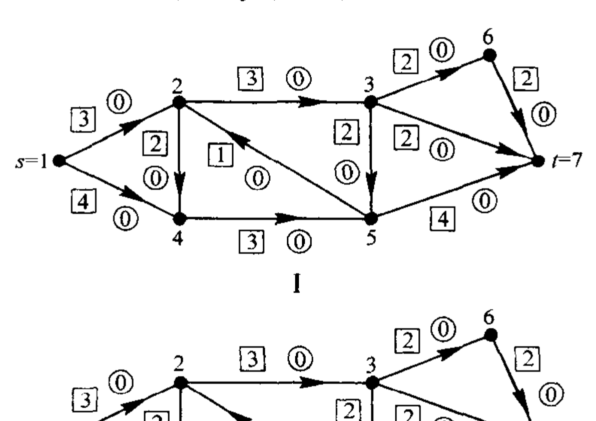
步骤 2.4. 通过把θ加入到弧(i, j)上的流来重新定义x，如果(i, j)在P中. 返回到步骤 2.1.

步骤 3. 停止于流x.

让我们把这一算法运用于图 13.10a 的有向网络. 图 13.12 给出由下面的迭代所定义的一系列流. 在第一次迭代中，对于所有的i，j，我们取$x_{ij} = 0$. 然后，我们注意到$P = 1, 4, 5, 7$是一个(s, t)路径，它的所有弧都是未饱和弧，且对应的θ等于 3，因为$s_{14} = c_{14} - x_{14} = 4 - 0 = 4$，$s_{45} = c_{45} - x_{45} = 3 - 0 = 3$，$s_{57} = c_{57} - x_{57} = 4 - 0 = 4$. 因此，我们使用$\theta = 3$增加$x_{14}$，$x_{45}$，$x_{57}$，得到图 13.12 中的第二个流. 在这个流中，$P = 1, 2, 3, 7$是一个(s, t)路径，它的所有弧都是未饱和弧. 它的$\theta = 2$，因为$s_{12} = 3$，$s_{23} = 3$，$s_{37} = 2$. 因此，我们使用$\theta = 2$增加x_{12}，x_{23}，x_{37}，得到图 13.12 中的第三个流. 在这个流中，$P = 1, 2, 3, 6, 7$是所有弧都未饱和的(s, t)路径，且$\theta = 1$，因为$s_{12} = c_{12} - x_{12} = 3 - 2 = 1$，$s_{23} = 1$，$s_{36} = 2$，$s_{67} = 2$. 因此，我们使用$\theta = 1$增加$x_{12}$，$x_{23}$，$x_{36}$，$x_{67}$，得到图 13.12 中的第四个流. 因为在这个流中，不再有所有弧都未饱和的(s, t)路径，所以我们停止. 注意我们得到一个值为 6 的流，我们知道这个流是最大流.

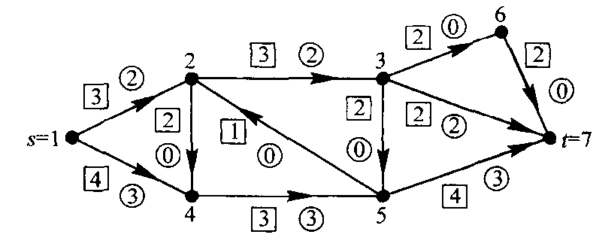
763

遗憾的是，这一算法不一定给出最大流. 我们考虑同一个有向网络. 图 13.13 给出使用不同的(s, t)路径在这一算法的另一次应用中的一系列步骤. 注意，在得到第四个流之后，我们无法找到所有弧都未饱和的(s, t)路径. 因此，这一算法停止. 然而，所得到的流的值等于 5，它不是最大流. 出现了什么错误？

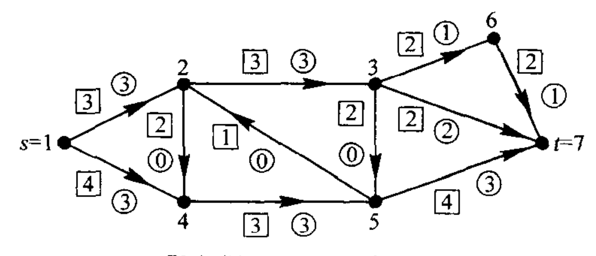


I

Ⅱ(根据 P=1,4,5,7及θ=3)



Ⅲ(根据 P=1,2,3,7及θ=2)



Ⅳ(根据 P=1,2,3,6,7及θ=1)

图 13.12 把算法 13.4 运用于图 13.10a 的有向网络

考虑最小分割 $S=\{1, 2, 4\}$，$T=\{3, 5, 6, 7\}$. 其中的一个问题是，所使用的一个单位流路径 1，4，5，2，3，7 是向后穿过这个分割的，即从 T 到 S；等价地，它从 S 到 T 向前流过这个分割两次. 因此，P 上的 1 个单位流用尽 2 个单位容量，这使用了过多的容量. 如果我们避开向后流动的某些流，并更多使用向前的流，那么我们的处境就会好转. 这给出改进这一算法的一个方法.

13.3.4　增广链

考虑忽略弧的方向从有向网络 D 得到图(多重图)G. 设 C 是这个图中从 s 到 t 的链. D 的弧 (i, j) 称为 C 的向前弧(forward arc)，如果它的流向是从 i 到 j. 否则称为向后弧(backward arc). 例如，在图 13.10 中的有向网络 D 中，一条链是 1，4，5，3，6，7. 这里，弧(1，4)是向前弧，而弧(3，5)是向后弧. 如果 x 是一个流，C 被称为一个相对于 x 的流增广链(flow-augmenting chain relative to x)，或简称为相对于 x 的增广链(augmenting chain relative to x)，如果对于每一个向前弧有 $x_{ij}<c_{ij}$，且对于每一个向后弧有 $x_{ij}>0$. 我们在 13.3.3 节讨论的没有饱和弧的单位流路径都对应于没有向后弧的流增广链. 图 13.10b 中的链 1，4，5，3，6，7 是相对于这个图中所示的流的流增广链. 因为向前弧(1，4)，(4，5)，(3，6)和(6，7)都小于容量，而向后弧(3，5)有正的流. 我们可以通过减小向后流并增加向前流来改进流的值. 特别地，我们对这个链上的所有向后弧记录 x_{ij}，并对所有向前弧计算并记录裕量 $s_{ij}=c_{ij}-x_{ij}$. 如果 λ 是这些记录中的最小值，那么称 λ 是这个增广链的容量(capacity). 于是，我们把向前弧上的每一个 x_{ij} 增加 λ 并把向后弧上的每一个 x_{ij} 减少 λ. 根据 λ 的选择，每一个新的 x_{ij} 仍然是非负的，而且不大于容量 c_{ij}. 另外，守恒定律(13.6)仍然成立. 接下来，我们观察到这个流的值增加 λ，因为这个链开始于从 s 出发的边并结束于到 t 的边. 如果从 s 出发的这条边是向前的，从 s 出来的流被增大；如果它是向后的，那么进入 s 的流被减小. 无论哪种情况，这个值，或从 s 出来的净流都被增加. 对于进入 t 的边，类似的讨论也成立[在流增广链不是简单链的情况下，这些讨论需要扩展. 其扩展留给读者(练习 35)]. 然而，我们只需简单流增广链(练习 36). 为了使用图 13.10b 的例子来进行说明，注意到 $s_{14}=2$，$s_{45}=1$，$s_{36}=1$，$s_{67}=1$，$x_{35}=2$. 因此，这些数的最小数 λ 为 1. 我们把 x_{14}，x_{45}，x_{36}，x_{67} 增加 1，并把 x_{35} 减小 1，得到一个值为 6 的流，它比所示的流的值大 1.

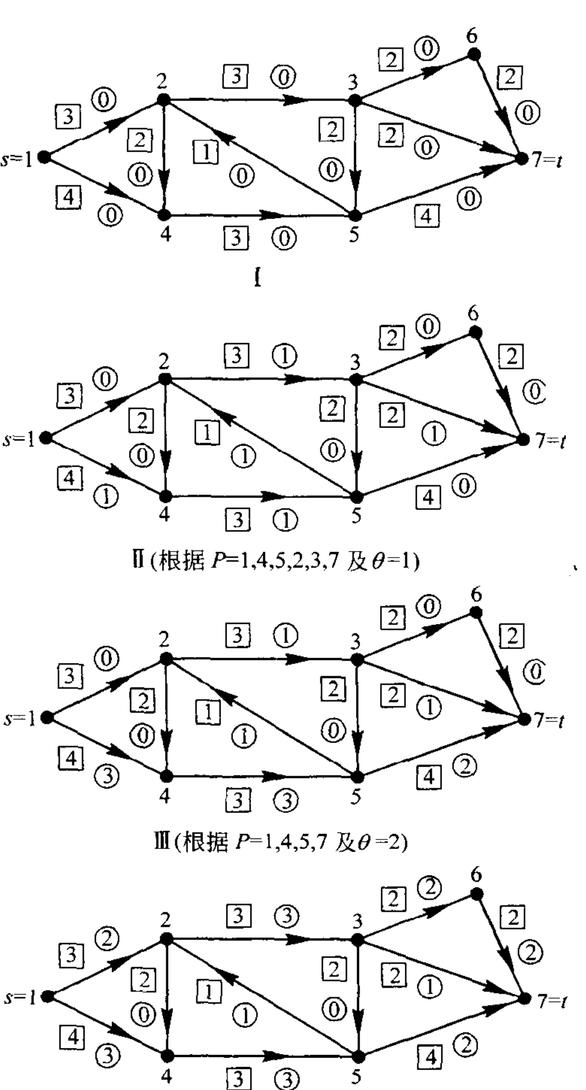


图 13.13　把算法 13.4 运用于图 13.10a 的有向网络，使用不同的(s，t)路径选择

764 ~ 766

现在我们已经做好了陈述关于最大流的主要结果的准备，这一结果使得我们能够给出我们

的主要算法. 我们已经证明，如果一个流有一个增广链，那么我们可以增大这个流的值，所以这个流不是最大的. 第一个结果说明，如果一个流不是最大的，那么我们可以寻找一个增广链(因此，流增广链类似于匹配中的M增广链，而下面的定理类似于定理12.7. 我们将在13.3.8节扩展网络流与匹配之间的关系).

定理 13.5　一个(s, t)流是最大的，当且仅当它没有从s到t的增广链.

证明㊀　还需假设x是一个没有增广链的流并证明x是最大的. 设S是存在从s到j的增广链的顶点j的集合，并设T是所有其他顶点的集合. 注意s在S中，因为单个s定义一个从s到s的增广链. 另外，t在T中，因为不存在增广链. 根据增广链的定义及S和T的定义，对于S中的所有i和T中的所有j我们有$x_{ij}=c_{ij}$，$x_{ji}=0$. 这是因为，由于i在S中，存在从s到i的增广链. 如果$x_{ij}<c_{ij}$或$x_{ji}>0$，那么我们可以把边$\{i, j\}$加到这条链中来寻找从s到j的增广链，这与j在T中矛盾. 因此，对于所有S中的i和T中的j，有$x_{ij}-x_{ji}=c_{ij}$.

根据推论13.3.1，

$$v=\sum_{i\in S}\sum_{j\in T}[x_{ij}-x_{ji}]=\sum_{i\in S}\sum_{j\in T}c_{ij}=c(S,T).$$

因此，我们找到一个容量与流x的值相同的分割(S, T). 根据定理13.4，这个流是最大的. ■

一个分割(S, T)称为相对于流x饱和的(saturated)，如果对于所有$i\in S$，$j\in T$，有$x_{ij}=c_{ij}$，且对于所有$i\in S$，$j\in T$有$x_{ji}=0$. 下面是定理13.5的证明的推论.

推论 13.5.1　如果(S, T)是相对于流x饱和的(s, t)分割，那么x是最大流.

767

下面的结果是归功于Elias，Feinstein和Shannon[1956]及Ford和Fulkerson[1956]的著名定理.

定理 13.6(最大流最小分割定理)　在一个有向网络中，(s, t)流的最大值等于(s, t)分割的最小容量.

证明㊁　根据定理13.3可知，(s, t)流的最大值至多等于(s, t)分割的最小容量. 为了证明二者相等，我们假设x是一个最大流，其值等于v. 那么它不可能有流增广链，因此，根据定理13.5的证明，我们能够寻找(s, t)分割(S, T)，使得$v=c(S, T)$. 根据定理13.4可得x是最大流而(S, T)是最小分割. ■

注释. 最大流最小分割定理的证明使用了最大流存在的隐含假设. 如果所有容量都是有理数，那么这一事实很容易证明. 因为13.3.5节所描述的最大流算法可以寻找到一个最大流. 如果某些容量不是有理数，最大流仍然存在(正如我们的说明，即使下面的最大流算法不一定能找到最大流). 关于其证明可以参见Lawler[1976]、Papadimitriou和Steiglitz[1982].

13.3.5　最大流算法

现在我们能够阐述最大流算法.

算法 13.5：最大流算法

输入：有源头s和汇聚t的有向网络.

输出：最大(s, t)流x.

步骤 1. 对于所有i，j，设置$x_{ij}=0$.

步骤 2.

步骤 2.1. 寻找一条从s到t的流增广链C. 如果不存在这样的增广链，则进入步骤3.

步骤 2.2. 计算并记录C中每一条向前弧的裕量并记录C的每一条向后弧的流.

㊀ 本证明可以跳过.

㊁ 本证明可以跳过.

步骤 2.3. 计算在步骤 2.2 中所记录的数中的最小值 λ.

步骤 2.4. 通过把 λ 加入到 C 的所有向前弧的流中并从 C 的所有向后弧的流中减去 λ 来重新定义 x. 返回到步骤 2.1.

步骤 3. 停止于流 x.

假设我们把这一算法运用于图 13.10a 中的有向网络. 因为每一条(s, t)路径都是没有向后弧的增广链，所以我们能够得到图 13.13 的第四个流. 然后，我们确定流增广链 $C=1, 2, 5, 7$. 在步骤 2.2，我们计算 $s_{12}=1, s_{57}=2, x_{52}=1$. 然后，得到这些值中的最小值 $\lambda=1$. 我们把 x_{12} 和 x_{57} 增加 $\lambda=1$，而把 x_{52} 减少 $\lambda=1$，得到图 13.14 的流. 在这个流中没有增广链. 我们得出这是最大流的结论. 这与我们前面的结论一样，因为这个流的值是 6.

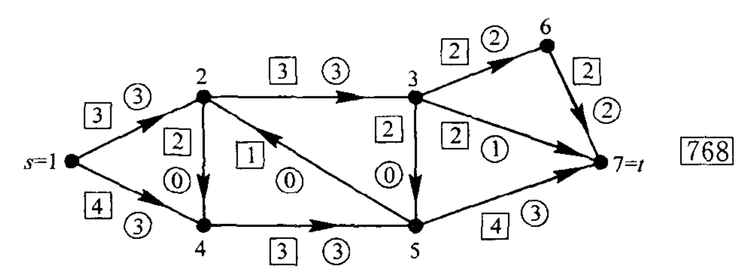


图 13.14 通过使用流增广链 $C=1, 2, 5, 7$ 从图 13.13 的第四个流得到的流

768

定理 13.7 *如果有向网络中的所有容量都是有理数，那么最大流算法将得到最大流.*

证明[⊖] 显然，如果所有容量都是整数，那么在每一次迭代中，数 λ 是一个整数，而且流的值增加 λ. 因为这个值至多是由 $S=\{s\}$，$T=V(D)-\{s\}$所定义的分割的容量，所以这个值递增的整数值不能超过有限数量次. 因此，将会出现不存在从 s 到 t 的增广链的情况. 根据定理 13.5，对应的流是最大的. 对于有理数容量的情况，我们可以通过下面的方法处理：对所有容量寻找一个公共分母 δ，并考虑由原来的有向网络通过把所有容量乘以 δ 而得到的新有向网络. ■

一般地，如果某些容量不是有理数，那么这个算法不一定在有限步数内收敛[⊜]. 另外，正如 Ford 和 Fulkdeson[1962]所指出的那样，它可以收敛于一个不是最大的流的值，Edmonds 和 Karp[1972]已证明，如果沿着一个有最小边数的增广链做每一个流增广，那么这一算法在有限步数内停止，且得到最大流. 其详细内容参见 Ahuja，Magnanti 和 Orlin[1993]、Lawler[1976]、Papadimitriou 和 Steiglitz[1982]. Ahuja，Magnanti 和 Orlin[1989，1991]概览了网络流算法的各种改进方法. 769

13.3.6 寻找增广链的标记过程[⊜]

我们所描述的最大流算法没有讨论如何在步骤 2.1 寻找增广链. 本小节描述寻找这样的链的一个标记过程(这个过程归功于 Ford 和 Fulkerson[1957]). 在这一过程的每一步，我们*扫描*(scan)一个顶点并给它的邻居标上标签. 顶点 j 得到标签(i^+)或(i^-). 我们通过寻找一个结束于边$\{i, j\}$的从 s 到 j 的增广链来决定标签. $+$表示$\{i, j\}$是这个增广链中的向前弧，而$-$表示它是一个向后弧. 最后，如果顶点 t 被标记，我们有一条从 s 到 t 的增广链. 如果这个过程得到没有标记顶点 t 的结论，那么不存在增广链，而且我们得出这个流最大的结论. 这个过程的详细描述如下.

算法 13.6：子程序——寻找增广链的标记算法

输入： 有源头 s 和汇聚 t 的有向网络，以及(s, t)流 x.

输出： 一个增广链或汇聚 x 是最大流的信息.

⊖ 本证明可以跳过.

⊜ 这一点没有实际意义，因为计算机是用有理数工作的.

⊜ 如果时间紧张，本小节可以跳过.

步骤 1. 给顶点 s 赋标签(−).

步骤 2. 设 F 是使得 $s_{ij}>0$ 的弧(i, j)的集合. 设 B 是使得 $x_{ij}>0$ 的弧(i, j)的集合(注意,F 中的弧可以用作增广链中的向前弧,而 B 中的弧可以作为增广链中的向后弧).

步骤 3.(标记和扫描)

步骤 3.1. 如果所有已标记的顶点被扫描,进入步骤 5.

步骤 3.2. 如果不是,寻找一个已标记但没有扫描的顶点 i,并如下扫描它.

对于每一条弧(i, j),如果$(i, j)\in F$ 而 j 没有被标记,给 j 赋标签(i^+).

对于每一条弧(j, i),如果$(j, i)\in B$ 且 j 没有被标记,给 j 赋标签(i^-).

不标记 i 的其他任意邻居. 顶点 i 现在已被扫描.

步骤 3.3. 如果顶点 t 已被标记,进入到步骤 4. 否则进入到步骤 3.1.

步骤 4. 从顶点 t 开始,使用索引标签构建一个增广链. 顶点 t 的标签表明这个链上倒数第二个顶点,倒数第二个顶点上的标签表明这个链上的它的前辈,以此类推. 停止并输出这个链.

步骤 5. 停止并输出流 x 是最大流的信息.

我们利用图 13.10b 的流来说明这个算法. 我们从把顶点 s 标记为(−)开始. 然后,我们发现 F 是由弧(1, 4),(2, 4),(3, 6),(3, 7),(4, 5),(5, 2)和(6, 7)组成的,而 B 是由除弧(2, 4),(5, 2),(3, 7)之外的所有弧组成的. 现在,我们进行标记和扫描过程(步骤 3). 图 13.15给出这些标签. 注意,顶点 $s=1$ 已标记但没有被扫描. F 中的唯一弧$(1, x)$是(1, 4). 因此,我们用(1^+)标记顶点 4. 在 B 中不存在弧$(x, 1)$. 因此,顶点 1 已被扫描. 因为顶点 $7=t$ 还没有被标记,所以我们寻找另一个已标记但没有被扫描的顶点 4. 现在,我们考虑 F 中的弧$(4, x)$,即弧(4, 5). 因此,顶点 5 得到标签(4^+). 我们也注意到弧$(x, 4)$不在 B 中. 因此,顶点 4 已被扫描. 注意,t 还没有被标记,所以我们寻找另一个已标记但没有被扫描的顶点 5. 而弧(5, 2)在 F 中. 所以我们给顶点 2 标记(5^+). 弧(3, 5)在 B 中,所以我们标记顶点 3 为(5^-). 顶点 5 已被扫描. 现在 t 还没有被标记. 我们寻找另一个已标记但没有被扫描的顶点. 我们有两个选择顶点 2 和 3. 假设我们选择顶点 3. 扫描顶点 3 导致在顶点 6 和 7 上有标签(3^+). 现在 $t=7$ 已被标记(很巧合,所有顶点都被标记). 我们进入到步骤 4,并向后浏览寻找流增广链. 特别地,顶点 7 上的标签(3^+)把我们送回到顶点 3. 标签(5^-)把我们送回到顶点 5,标签(4^+)把我们送回到顶点 4,标签(1^+)把我们送回到顶点 1. 因此,我们有一条流增广链 1, 4, 5, 3, 7. 在这个例子中,为了寻找流增广链我们不必区分+和−. 然而,如果顶点 3 和 5 之间有两条弧(3, 5)和(5, 3),那么顶点 3 上的标签 5^- 会告诉我们使用向后弧(3, 5). 为了计算最大流算法所需的数 λ,在修改这一标记算法时,标签+和−很有用.

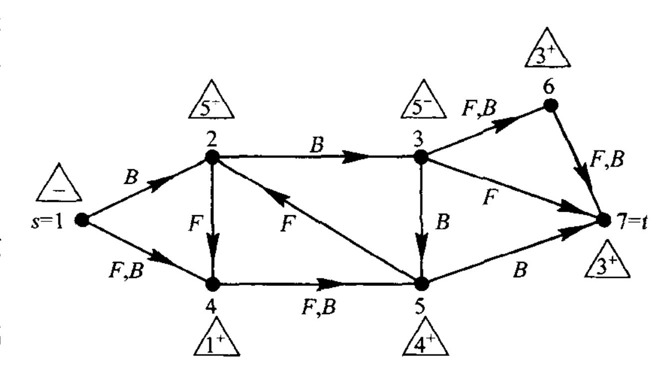


图 13.15 运用于图 13.10b 中的流的标记算法. 标签如顶点旁边的三角形中的符号所示. F 中的弧和 B 中的弧分别标记为 F 和 B

定理 13.8 如果流 x 不是最大的,那么标记算法寻找一个增广链,否则它以得出 x 是最大的结论而结束.

770

证明[1]　显然，如果这个算法产生一条从 s 到 t 的链，那么这条链是增广的. 我们证明如果 t 还没有被标记，且不存在已标记又没有被扫描的顶点，那么这个流是最大的. 设 S 是由所有已标记顶点组成的，而 T 是由所有未标记顶点组成的. 那么 (S, T) 是一个 (s, t) 分割. 而且，从 S 中的 i 到 T 中的 j 的每一条弧都是饱和的，且从 T 中的 j 到 S 中的 i 的每一条弧有流 0，因为否则我们就能够在扫描 S 的顶点时标记 T 中的一个顶点. 我们得出 (S, T) 是一个饱和分割的结论. 根据推论 13.5.1，我们得出 x 是最大流的结论. ■

[771]

在结束本小节时，我们注意到可以修改标记算法，使得在结束时容易计算最大流算法所需的数 λ. 在给一个顶点 j 指定标签的每一步，我们刚好寻找到一条从 s 到 j 的增广链. 于是，设 $\lambda(j)$ 是这条链中的向前弧 (u, v) 的数量 s_{uv} 和这条链中的向后弧 (u, v) 的数量 x_{uv} 中最小的数. 令 $\lambda(s)$ 是 $+\infty$. 如果通过从 i 开始扫描来标记 j，那么可以根据 $\lambda(i)$ 计算 $\lambda(j)$. 特别地，如果 j 得到标签 (i^+)，那么 $\lambda(j)=\min\{\lambda(i), s_{ij}\}$，如果 j 得到标签 (i^-)，那么 $\lambda(j)=\min\{\lambda(i), x_{ji}\}$. 最后，$\lambda$ 等于 $\lambda(t)$（参见练习 40). 为了说明，在我们的例子中，可以按下面的顺序计算 $\lambda(j)$：$\lambda(s)=+\infty$，$\lambda(4)=\min\{\lambda(s), s_{14}\}=2$，$\lambda(5)=\min\{\lambda(4), s_{45}\}=1$，$\lambda(2)=\min\{\lambda(5), s_{52}\}=1$，$\lambda(3)=\min\{\lambda(5), x_{35}\}=1$，$\lambda(6)=\min\{\lambda(3), s_{36}\}=1$，$\lambda(7)=\min\{\lambda(3), s_{37}\}=1$，并且得出 $\lambda=\lambda(7)=1$ 的结论.

13.3.7　最大流算法的复杂度

我们对最大流算法的计算复杂度做一下评估. 设 a 是有向网络中弧的数量，而 v 是最大流. 13.3.6 节所描述的标记算法要求在每一次用于寻找增广链时至多做 $2a$ 次弧检查，作为向前弧，我们检查每一条弧至多一次，而作为向后弧我们检查每一条弧至多一次. 如果所有容量都是整数，而且对于所有 i, j 我们从流 $x_{ij}=0$ 开始最大流算法，那么每一条流增广链增加的值至少为 1，所以涉及搜索增广链的迭代数量至多是 v. 因此，所需的步骤数量至多是 $2av$[2]. 这一计算存在的一个问题是，我们希望把复杂度完全确定为输入大小的函数，而不是用解 v 表示的函数. 注意，我们的计算意味着这一算法也许花很长时间. 实际上，它可能花费很长时间. 考虑图 13.16 中的有向网络. 从 0 流开始，我们可能想到选择第一条增广链 1，2，3，4，然后选择增广链 1，3，2，4，再然后选择 1，2，3，4，最后是 1，3，2，4 等. 在收敛之前这将需要 20 亿次迭代！然而，如果第一次我们恰好选择了增广链 1，2，4，然后是增广链 1，3，4，那么我们用两次迭代就可以完成！为了避开这类问题，我们把标记算法的步骤 3.2 做一下修改，使得顶点的扫描顺序与它们得到标签的顺序一致. Edmonds 和 Karp[1972]已证明在这种情况下，至多运用 $an/2$ 次标记算法就可以得到最大流，其中 n 是顶点数量. 因此，因为每一次使用标记算法至多需要 $2a$ 步，所以总步数[3]至多是

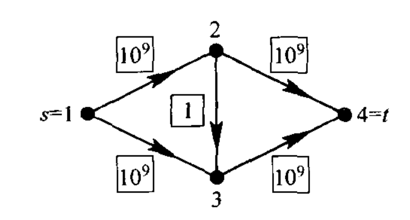


图 13.16　导致最大流算法进行 20 亿迭代的糟糕的增广链选择

[772]

$$(2a)(an/2) = a^2 n \leqslant [n(n-1)]^2 n.$$

---

[1] 本证明可以跳过.

[2] 这一结果忽略了计算裕量和修改流的简单计算. 在每一次迭代中，对于每一条弧，至多使用两次计算，所以要将步骤数加上一个常数倍 av. 在每一次迭代中，我们还需要计算 λ，它是不超过 a 个数的集合的最小数. 计算所有 λ 的总计算数量因此又是至多等于 av. 最后，我们必须根据标签构建增广链，在每一次迭代中，这一工作又是至多花费 a 步，或总共至多花费 av 步. 总之，如果 k 是个常数，那么利用 2.18 节的标记法，步骤总量因此至多是 kav，即 $O(av)$.

[3] 正如本页的脚注[2]所述，更精确的计算是至多需要 $ka(an/2)$ 步.

Edmonds 和 Karp 算法后来得到了改进. 有关改进算法的内容可以参见 Ahuja, Magnanti 和 Orlin[1993]、Papadimitriou 和 Steiglitz[1982].

13.3.8 再论匹配[⊖]

本小节研究网络流与我们在第 12 章中学习的匹配之间的关系. 特别地, 我们要说明如何证明定理 12.5, 即在二部图中, 最大匹配中的边的数量等于最小覆盖中的顶点的数量.

假设 $G=(X, Y, E)$ 是二部图. 我们可以把它转换成一个有向网络 D, 称为关联网络 (associated network), 其方法是加入源头 s 和汇聚 t, 从 s 到 X 中所有顶点的弧, 及从 Y 中所有顶点到 t 的弧, 同时定向 X 和 Y 之间的所有边的方向为从 X 到 Y. 在所有新弧上放置容量 1, 在所有从 X 到 Y 的弧上放置容量 ∞ (或者一个非常大的数). 于是, 容易看到 D 中从 s 到 t 的任意整数流 (所有 x_{ij} 值都是整数的流) 对应于 G 中的匹配 M: 对于 $i\in X$, $j\in Y$, 我们把边 $\{i, j\}$ 包含进 M 中, 当且仅当沿着弧 (i, j) 存在一条正流. 相反, G 中的任意匹配 M 定义 D 中的整数流 x: 对于 E 中的 $\{i, j\}$, 其中 i 在 x 中, 当 $\{i, j\}$ 在 M 中时取 x_{ij} 为 1, 否则取 x_{ij} 为 0; 当且仅当 k 在 M 中饱和时取 x_{sk} 为 1, 而当且仅当 l 在 M 中饱和时取 x_{lt} 为 1. 另外, 匹配 M 中的边的数量等于流 x 的值. 图 13.17 通过给出二部图 G 及其关联网络 D 说明了这一构造法. D 中所示的流对应于 G 中波状边所示的匹配. 我们用下面的定理概括这些结果.

773

图 13.17 二部图 G 及其关联网络 D. 以圆圈中的数所示的 D 中的流对应于波状边所示的 G 中的匹配. 容量为方格中所示的数

定理 13.9 *假设 $G=(X, Y, E)$ 是二部图, D 是它的关联网络. 那么 D 中的整数 (s, t) 流与 G 中的匹配之间存在一一对应. 另外, 整数流的值等于所对应的匹配中的边数.*

因为二部图 $G=(X, Y, E)$ 中的匹配对应于它的关联网络 D 中的流, 所以我们也许要问这个二部图中的覆盖对应于网络中的什么? 答案是覆盖对应于有限容量的 (s, t) 分割. 为了使这一讨论更精确, 假设 $A\subseteq X$, $B\subseteq Y$. 对于这样的 A 和 B, G 中的任意覆盖可以写成 $A\cup B$ 的形式表示. D 的任意一个 (s, t) 分割 (S, T) 可以写成下面的形式:

$$S=\{s\}\cup(X-A)\cup B, \quad T=\{t\}\cup A\cup(Y-B);$$

其中 s 属于 S, t 属于 T. 我们简单地把 A 定义为 $T\cap X$ 中的所有顶点, 而 B 是 $S\cap Y$ 中的所有顶点 (参见图 13.18). 现在, 我们有下面的定理.

定理 13.10 *假设 $G=(X, Y, E)$ 是二部图, 而 D 是其关联网络. 假设 $A\subseteq X$, $B\subseteq Y$. 设 $K=A\cup B$, 并设*

$$S=\{s\}\cup(X-A)\cup B, \quad T=\{t\}\cup A\cup(Y-B). \tag{13.12}$$

⊖ 本小节可以跳过.

那么 K 是 G 的覆盖，当且仅当 (S, T) 是 D 的有限容量 (s, t) 分割．另外，如果 (S, T) 是有限容量 (s, t) 分割，那么 $|A \cup B|$ 是这个分割 (S, T) 的容量．

证明　假设由(13.12)式所定义的 (S, T) 是有限容量 (s, t) 分割．因为这个分割的容量是有限的，所以在这个分割中不存在从 X 到 Y 的弧，即不存在从 $X-A$ 到 $Y-B$ 的弧(参见图 13.18)．因此，根据(13.12)式，D 中所有从 X 到 Y 的弧或者是从 A 出发的弧，或者是到 B 的弧，因此 G 的所有边都与 A 连接或与 B 连接，所以 $K=A \cup B$ 是一个覆盖．接下来，注意，$|A \cup B|$ 是分割 (S, T) 的容量．因为根据图 13.18，这个分割中的弧正好是 A 中 x 的弧 (s, x) 和 B 中 y 的弧 (y, t)．存在 $|A \cup B|$ 条这样的弧．每一条弧都有单位容量，所以这个分割有容量 $|A \cup B|$．

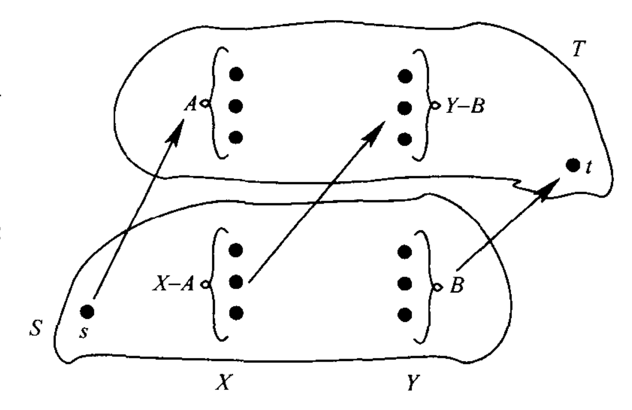


图 13.18　分割(13.12)式．分割中最好的可能弧由箭头表示

剩下要证明，如果 $K=A \cup B$ 是 G 的覆盖，那么由(13.12)式所定义的 (S, T) 给出一个 (s, t) 分割，它有有限容量．这一证明留给读者(练习 37)．■

现在我们证明定理 12.5.

定理 12.5 的证明[定理 12.5：假设 $G=(X, Y, E)$ 是二部图．那么最大匹配中的边的数量等于最小覆盖中的顶点的数量．]　考虑 G 的关联网络．因为所有容量都是整数，所以根据定理 13.7 的证明可得，存在一个最大流，在这个最大流中所有的 x_{ij} 都是整数．这个流的值也一定是一个整数，而且根据定理 13.9，这个流的值将给出最大匹配中的边数 α．另外，根据最大流最小分割定理(定理 13.6)，这个数 α 也是一个 (s, t) 分割 (S, T) 的最小容量．假设这个分割是由(13.12)式给出的．此时，显然，最小容量的 (s, t) 分割有有限容量，所以根据定理 13.10，通过(13.12)式分割 (S, T) 对应于一个覆盖 $K=A \cup B$．K 中的顶点数量等于 $|A \cup B|$，这个值等于分割 (S, T) 的容量，即等于 α．于是，存在有 α 个顶点的覆盖，和有 α 条边的匹配，所以根据定理 12.4，这个覆盖一定是最小的．因此，最大匹配中的边数等于最小覆盖中的顶点数量．■

774 ~ 775

13.3.9　门格定理[⊖]

最大流最小分割定理在通信网络的设计中非常有用．在设计这样的网络时，我们希望确保当某些连通被破坏之后，它们的“连通性”仍然保持完整，因此我们的兴趣就在于弧的删除能破坏从顶点 a 到顶点 z 的所有路径的弧的最小数量．保证连通性的一个方法是，确保我们嵌入“冗余”，并在顶点对之间有很多路径．我们称从 a 到 z 的一对路径是弧不相交的(arc-disjoint)，如果它们没有任何公共弧．我们将看到，弧删除后破坏从 a 到 z 的所有(简单)路径的弧的最小数量与从 a 到 z 的弧不相交简单路径的最大数量之间有着密切的关系．下面的结果就是称为门格定理(以 Menger[1927]的名字命名)的一系列重要定理之一．

定理 13.11　*设 N 是以 a 为源头、以 z 为汇聚的连通有向网络，网络中每一条弧的容量为 1．那么：*

(a) N 中的最大 (a, z) 流的值等于从 a 到 z 的弧不相交简单路径的最大数量 M．

(b) N 中的最小 (a, z) 分割的容量等于弧删除后破坏从 a 到 z 的所有简单路径的弧的最小

⊖　本小节可以跳过．

数 p.

证明 我们给出 a 只有出去的弧而 z 只有进入弧的情况的证明. 一般情况的证明留给读者(练习 32).

(a)设 F 是所有整数 k 的集合，使得如果在容量都为 1 的有向网络中存在值为 k 的(a, z)流，那么存在从 a 到 z 的 k 个弧不相交的简单路径的集合. 显然，$k=1$ 在 F 中. 现在我们通过对 k 使用归纳法证明，所有整数都在 F 中. 假设 k 在 F 中，我们证明 $k+1$ 在 F 中. 假设存在值为 $k+1$ 的流. 这时，存在从 a 到 z 的简单路径 P，其中的每一条弧都有非零流. 因为所有容量都为 1，所以这个流在 P 的每一条弧上都是 1. 剩余的 k 个单位流一定经过不在 P 上的弧. 因此，删除 P 的这些弧，留下一个有向网络，它的容量都为 1，而且有值为 k 的流. 根据归纳假设，在这个新的有向网络中，存在 k 条从 a 到 z 的弧不相交简单路径. 这些路径连同 P 形成原来有向网络中的 $k+1$ 条从 a 到 z 的弧不相交简单路径的集合. 这就证明如果 v 是最大(a, z)流的值，那么 $v \leqslant M$.

假设我们在 N 中有 M 条从 a 到 z 的弧不相交简单路径. 如果我们定义 x_{ij} 在这样一条路径中的所有弧(i, j)上等于 1，否则 x_{ij} 等于 0，那么显然 $x=\{x_{ij}\}$是一个值为 M 的流，所以 $v \geqslant M$.

(b)设(S, T)是 N 中的一个(a, z)分割. 这时，如果从 S 到 T 的所有弧都被破坏，那么 z 是从 a 不可达的. 因此，这个分割的容量至少为 p.

现在，设 A 是弧删除后破坏从 a 到 z 的所有简单路径的 p 条弧的集合，设 N' 是通过删除 A 的弧从 N 得到的有向网络. 设 B 是 N' 中从 a 可达的所有顶点的集合. 根据 B 的定义，不存在从 B 的顶点到 $B'=V(N)-B$ 的顶点的弧，所以在 N 中从 B 到 B' 的所有弧都在 A 中. 因此有(B, B')是容量至多等于 p 的分割，从而它一定是最小分割. ■

[776]

定理 13.12(Menger[1927]) *在有向图 D 中，从顶点 a 到 z 的弧不相交简单路径的最大数量等于 D 中弧删除后破坏从 a 到 z 的所有简单路径的弧的最小数量.*

证明 通过给 D 的每一条弧分配单位容量来构建一个源头为 a，下落为 z 的有向网络. 根据定理 13.11 和最大流最小分割定理(定理 13.6)，此定理成立. ■

为了说明这一定理，考虑图 13.10(忽略容量)的有向网络. 这时，存在两条从 s 到 t 的弧不相交简单路径：即 s, 2, 3, 6, t 和 s, 4, 5, t. 同时，存在这样的两条弧，删除它们将破坏所有从 s 到 t 的(简单)路径：弧(2, 3)和弧(4, 5). 因此，可得从 s 到 t 的弧不相交简单路径的最大数量为 2，而删除后破坏从 s 到 t 的所有(简单)路径的弧的最小数量也是 2.

我们说一个有向图中从 a 到 z 的两条路径是*顶点不相交的*(vertex-disjoint)，如果它们除 a 和 z 之外没有公共顶点.

定理 13.13(Menger[1927]) *假设在有向图 D 中不存在从顶点 a 到 z 的弧. 那么 D 中从 a 到 z 的顶点不相交简单路径的最大数量等于 D 中顶点删除后破坏从 a 到 z 的所有简单路径的顶点的最小数量.*

证明 用下面的方法定义一个新的有向图 D'：把除 a 和 z 之外的每一个顶点 u 分裂成两个新顶点 u_1 和 u_2，加入弧(u_1, u_2)，用弧(u_2, w_1)取代弧(u, w)，用(u_2, z)取代弧(u, z)，用(a, u_1)取代弧(a, u). 这时，我们可以证明，在 D' 中从 a 到 z 的弧不相交简单路径的最大数量等于 D 中从 a 到 z 的顶点不相交简单路径的最大数量，而且弧删除后破坏 D' 中从 a 到 z 的所有简单路径的弧的最小数量等于顶点删除后破坏 D 中从 a 到 z 的所有简单路径的顶点的最小数量. 详细证明留作练习[练习 33(a)和(b)]. ■

为了说明这一定理，再次考虑图 13.10 的有向网络. 上面给出的从 s 到 t 的两条弧不相交简单路径也是顶点不相交的. 顶点删除后破坏从 s 到 t 的所有简单路径的两个顶点是顶点 2 和 4.

本节练习

1. 在图 13.19 的每一个有向网络中，圆圈中的数字给出一个(可能的)(s, t)流.

   (a)这些流中哪些是可行的？为什么？

   (b)如果这个流是可行的，那么确定它的值. [777]

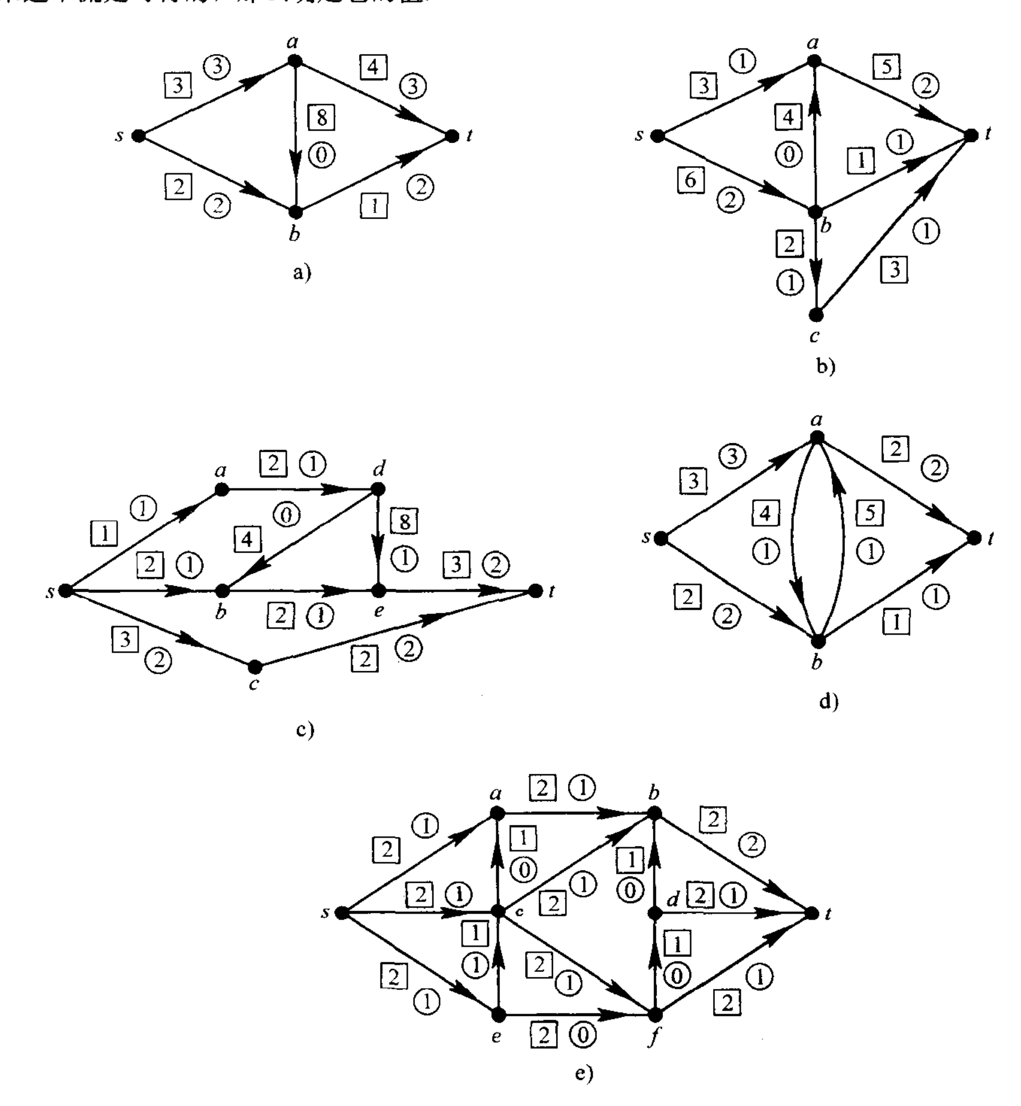


图 13.19　13.3 节练习用的(可能)流 [778]

2. 对于图 13.18 中的每一个有向网络 D，把权解释成容量，并寻找分割(S, T)的容量，其中 $S=\{a, b, e\}$，$T=V(D)-S$.
3. 对于图 13.20 中的每一个流，计算每条弧的裕量.
4. 对于图 13.13 中的流 III，下面中哪些是流增广链？

   (a)1，2，3，7　　(b)1，2，5，7　　(c)1，4，5，7　　(d)1，4，2，3，7
5. 对于图 13.12 中的流 III，练习 4 中的哪些链是流增广链？
6. 对于图 13.20 中的每一个流，或者通过寻找一个使得 $v=c(S, T)$的(s, t)分割(S, T)来证明它是最大的，或者通过寻找一个增广链证明它不是最大的.
7. 给出一个有向网络和这一网络的流 x 的例子，对于这个网络，流 x 有值 0，但是对于所有的 i，j，x_{ij} 不等于 0.

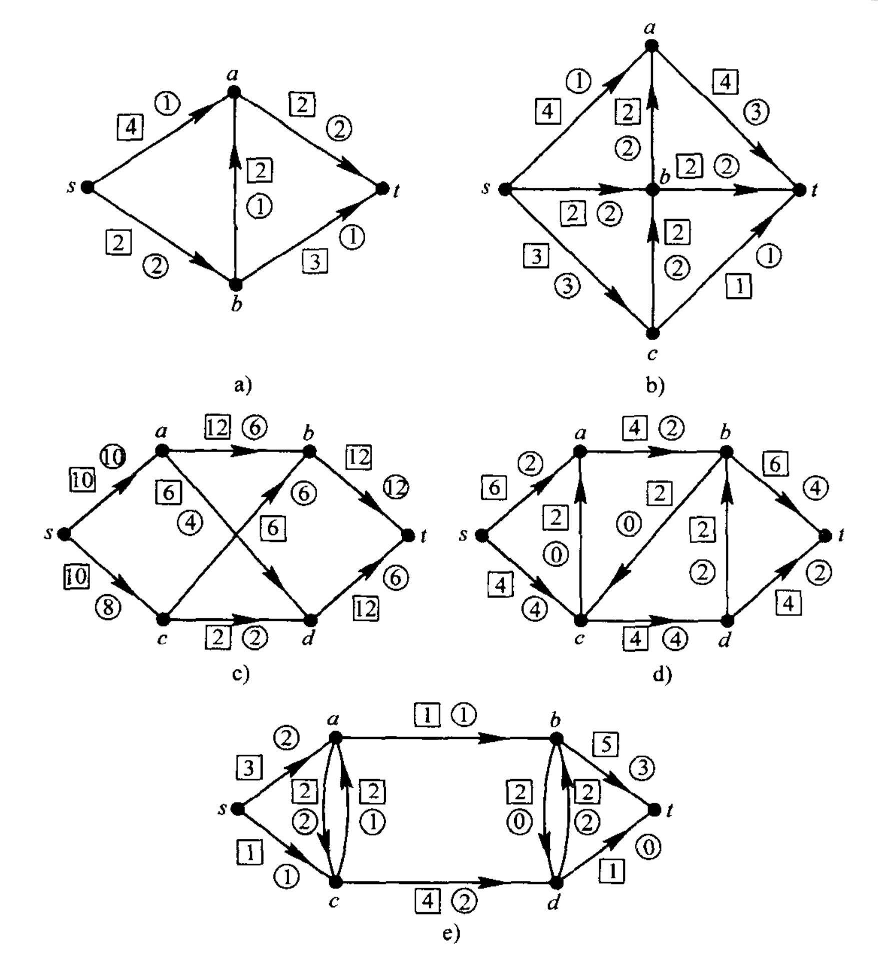


图 13.20 13.3 节和 13.4 节练习用的流

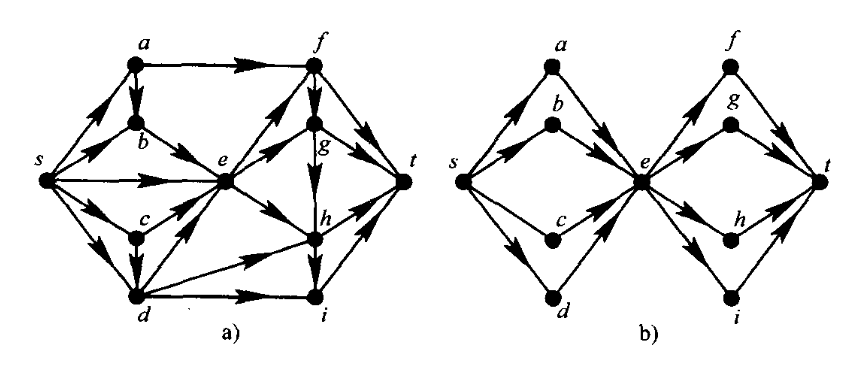


图 13.21 13.3 节练习用的有向图

8. 关于下面各图中的有向网络，说明门格(第一)定理——定理 13.12.

(a)图 13.19　　　　(b)图 13.21

9. 关于下面各图中的有向网络，说明门格(第二)定理——定理 13.13.

(a)13.19　　　　(b)图 13.21

10. 设 G 是无向图.

(a)陈述并证明门格(第一)定理的变形——定理 13.12.

(b)陈述并证明门格(第二)定理的变形——定理 13.13.

11. 把算法 13.4 运用于图 13.8 中的每一个有向网络，其中权被解释为容量，且 $s=a$，$t=z$.

12. 运用算法 13.5 重复练习 11.
13. 对于图 13.20 中的每一个流，运用算法 13.6 搜索增广链.
14. 对于练习 13 中算法 13.6 的每一个应用，计算数 $\lambda(j)$.
15. 在图 13.8 的有向网络(d)中，设 $s=a$，$t=z$，并通过设 $x_{ab}=x_{bc}=x_{cd}=x_{dz}=2$，$x_{ae}=x_{ef}=x_{fg}=x_{gz}=3$，且 $x_{ah}=x_{hi}=x_{ij}=x_{jz}=2$，而取其他的 x_{ij} 为 0 定义一个流. 运用算法 13.6 搜索流增广链.
16. 设 G 是一个图，S 和 T 是顶点的不相交子集. 证明一端在 S 中而另一端在 T 中的顶点不相交简单路径的最大数量等于满足下面条件的顶点的最小数量：删除这些顶点分离 S 和 T，即删除这些顶点后，不存在既包含 S 的顶点又包含 T 的顶点的连通分支.
17. 对于图 12.12 中的每一个二部图：
    (a)寻找其关联网络 D.
    (b)在 D 中寻找一个整数(s, t)流及其在 G 中的对应匹配.
    (c)在 D 中寻找一个(s, t)分割及其在 G 中的对应覆盖.
18. 考虑例 13.8 的学校招生问题. 下面的表格包含 8 个申请人的列表，列表包括他们的候选专业以及他们的所在州.

申请人	可能专业				所在州
	化学	生物学	数学	地理学	
A	x				MD
B		x			MD
C			x	x	NY
D	x	x			MD
E			x	x	PA
F			x		NJ
G			x		NY
H				x	PA

    (a)画出关联有向网络.
    (b)如果这所学校从每个州至多要 2 名学生，那么招生官员能够完成他们的工作吗?
    (c)如果这所学校从每个州至多要 1 名学生，那么招生官员能够完成他们的工作吗?
19. 考虑例 13.9 中双处理器计算机上的分布式计算问题. 表 13.4 给出在这两个处理器上每一个模块的执行成本，而表 13.5 给出处理器间的通信成本.

表 13.4　执行成本

i	1	2	3	4	5
a_i	6	5	10	4	8
b_i	3	6	5	10	4

表 13.5　处理器间的通信成本

$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}
\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 4 & 1 & 0 & 0 \\ 4 & 0 & 5 & 0 & 0 \\ 1 & 5 & 0 & 6 & 2 \\ 0 & 0 & 6 & 0 & 1 \\ 0 & 0 & 2 & 1 & 0 \end{bmatrix} \end{array}
$$

    (a)画出关联网络.
    (b)使用最小成本分割寻找处理器最小成本模块分配.
20. 回想一下，在无向网络中，从源头 s 到汇聚 t 的流被定义为通过下面方法而得到的有向网络中从 s 到 t 的流：用弧(i, j)和(j, i)取代每一条边$\{i, j\}$，并设每一条弧与对应的边有相同的容量. 对于图 13.22 中的每一个网络，寻找从 s 到 t 的最大流.
21. 在一个管道网络中，从位置 A 向位置 B 传输油. 油可以流经北线或南线的管道. 每一个管道有一个接点，其中一条管道是从南线接点到北线接点. 从位置 A 到北线管道接点的第一条支线的容量是每小时 400 桶油；而从这个接点到位置 B 的第二条支线的容量是每小时 300 桶. 南线管道的第一条支线的容量是每小时 500 桶，而第二条支线的容量是每小时 300 桶. 连接两个接点的管道的容量也是每小时 300 桶. 每小时从位置 A 到 B 所能传输的最多油量是多少桶?

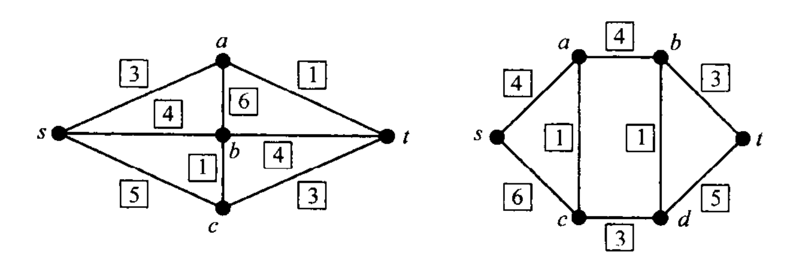


图 13.22 13.3 节练习用的网络

22. 在本练习中，我们以例 2.21 和例 3.10 中所讨论的系统可靠性的概念为基础. 假设一个系统是用有向网络表示的，其组件对应于弧. 我们说这个系统可行，当且仅当在以可行组件定义的修改网络中，存在值至少为 v 的(s, t)流. 对于图 13.23 中的每一个有向网络，如果 v 为 3，计算例 2.21 中定义的 $F(x_1x_2\cdots x_n)$.

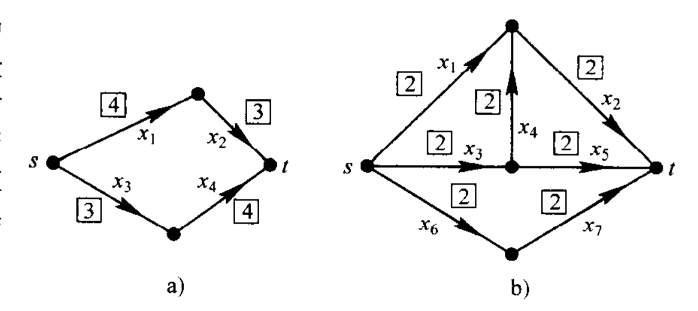


图 13.23 13.3 节练习用的有向网络

23. 假设 D 是有向网络，X 是源头的集合，Y 是汇聚的集合[我们假设 $X, Y \subseteq V(D)$，且 $X \cap Y = \varnothing$]. (X, Y)流是这样的流，其中守恒条件(13.6)只对非源头和非汇聚的顶点成立. 这个流的值由下式定义：

$$\sum_{\substack{i \in X \\ j \notin X}} x_{ij} - \sum_{\substack{i \in X \\ j \notin X}} x_{ji}.$$

我们可以如下寻找到最大(X, Y)流：在 D 中加入两个新顶点 s 和 t，对于 X 中的所有顶点，加入从 s 到这些顶点容量为∞的弧，对于 Y 的所有顶点，加入这些顶点到 t 容量为∞的弧，并在这个新网络中寻找最大(s, t)流. 对于图 13.24 中的每一个有向网络，寻找最大(X, Y)流.

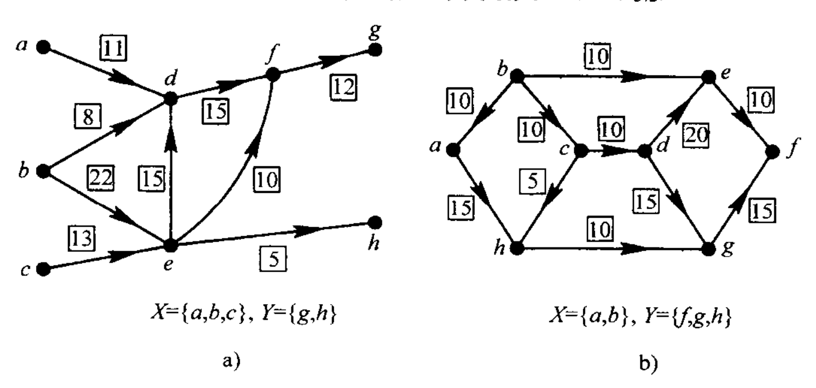


图 13.24 源头集合为 X、汇聚集合为 Y 的有向网络

24. 有 3 个仓库 w_1，w_2 和 w_3，和 3 家零售商店 r_1，r_2 和 r_3. 3 个仓库分别有 3000，4000，6000 桶油漆，而零售店分别需要 2000，4000，3000 桶油漆. 图 13.25 给出一个货运网络，其中弧(i, j)的容量给出在给定的一天内，从位置 i 到 j 运输油漆的最大桶数. 如果仅有一天允许从仓库运出货物，能否满足所有需要？(在一天中，一桶油漆可以经过多条必要的弧.)如果不能满足所有要求，那么能够满足的最大总需求是多少？我们可以把这一问题转化成多源头和多下落的问题，然后再把它转化成普通网络流问题(参

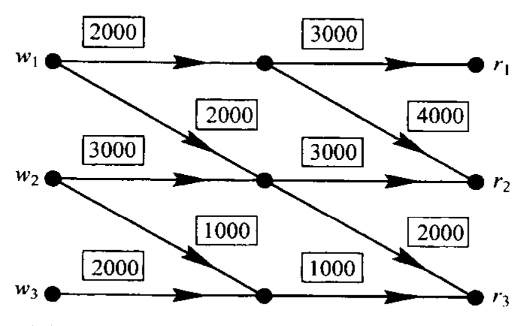


图 13.25 13.3 节练习 24 用的运输网络

见练习 23)，这样我们就可以解决这一问题. 这是一个*转载问题*(transshipment problem)的例子.

25. (Ahuja，Magnanti 和 Orlin[1993])一个单位有一定数量的技术工人，他们有不同的技能和不同的资历等级. 每一个人有指定技能集合中的至少一项技能，但仅有一个资历等级. 这个单位要组建一个适应某些条件的理事会：每一种类型技能的人有一个理事会成员，拥有资历等级 k 的成员至多是 u_k. 是否有可能寻找到这样的理事会？证明：答案是"是"当且仅当某个(s, t)有向网络有等于指定技能数量的最大流值.
26. 没有相同成员的若干个立法委员会要集会相互交流想法. 我们想把它们分配到不同的房间，使得同一委员会中没有两个人去相同的房间. 说明如何使用最大流公式为委员会成员分配房间. 你应该使用每一个委员会的成员数量以及第 i 个会议室的容量作为公式的输入.
27. 在某些有向网络中，我们也许有顶点容量和弧容量. 例如，对经过某一个给定的收费设施的汽车数量有一定的容量限制. 这时，我们寻找既满足弧容量限制又满足顶点容量限制的最大流. 把这一问题转化为只有弧容量限制的标准最大流问题.
28. 证明：如果 x 是值为 v 的(s, t)流，且(S, T)是容量为 c 的(s, t)分割，那么 $v=c$ 当且仅当对从 T 到 S 的每一条弧(i, j)有 $x_{ij}=0$，而对从 S 到 T 的每一条弧(i, j)有 $x_{ij}=c_{ij}$.
29. 假设从 s 到 t 的流被分解成从 s 到 t 的单位流路径，而且每一条单位流路径都正好穿过给定的饱和分割一次. 证明这个流是最大的.
30. 我们希望在某个区域把信使从位置 s 送到位置 t，这个区域的道路网络被模型化为一个有向图. 因为某些道路可能被封锁，所以我们希望每一位信使沿着一条与所有其他信使的路径完全不相交的路径行驶. 我们如何寻找派送信使的最大数量？[*提示*：使用单位容量].
31. 对于例 13.8，证明：

    (a)如果最大流的值等于代表申请人感兴趣的专业的数量，那么它可以解决"学校招生"问题.

    (b)如果上面的条件不成立，这一问题没有解.

779 ～ 784

32. 对于顶点 a 有进入弧，或顶点 z 有出去弧的情况，完成定理 13.11 的证明.
33. 这个练习给出定理 13.13 的证明. 假设在有向图 D 中不存在从顶点 a 到顶点 z 的弧. 如该定理"证明"中那样定义 D'.

    (a)证明 D'中从 a 到 z 的弧不相交简单路径的最大数量，等于 D 中从 a 到 z 的顶点不相交简单路径的最大数量.

    (b)证明 D'中弧删除后破坏所有从 a 到 z 的(简单)路径的弧的最小数量，等于 D 中顶点删除后破坏所有从 a 到 z 的简单路径的顶点的最小数量.
34. 设 N 是无向网络，其基本图是连通的. 回想一下，N 中的*分割*是把顶点划分成两个集合 S 和 T，或等价地，是将 S 的顶点和 T 的顶点连接起来的所有边的集合. 如果 s 和 t 分别是 N 的源头和下落，那么当 $s\in S$，$t\in T$ 时，(S, T)是(s, t)分割. 设 F 是 N 的边的集合.

    (a)证明：如果在 13.1 节练习 18 的意义下，F 是简单分割集合，那么它是分割.

    (b)证明：如果 F 是有最小容量的(s, t)分割，那么 F 是简单分割集合.
35. 假设 C 是流增广链，而 λ 是 C 的容量，其定义如 13.3.4 节所述. 如果我们把向前弧上的每一个 x_{ij} 增加 λ，而把向后弧上的每一个 x_{ij} 减小 λ，证明即使 C 不是简单链，这个值或从 s 出来的净流仍是增加的.
36. 证明：在最大流算法中，寻找简单链的流增广链就足够了.
37. 完成定理 13.10 的证明.
38. 使用 13.3.8 节的结果，在不先证明定理 12.5 的情况下，证明 König-Egerváry 定理(推论 12.5.1).
39. 假设 $G=(X, Y, E)$是二部图，M 是 G 的匹配. 使用 13.3.8 节的结果证明定理 12.7，即 M 是最大的当且仅当 G 不含 M 增广链.
40. 证明：在标记算法中，如果 $\lambda(j)$的定义如定理 13.8 的证明之后的讨论，那么 $\lambda=\lambda(t)$.

13.4 最小成本流问题

一些例子

另一个网络流问题如下面所述，它有很多重要的特殊情况. 假设在一个有向网络中，除每一条弧上有非负的容量 c_{ij} 外，我们还有从 i 到 j 运送一个单位流的非负成本 a_{ij}. 我们要寻找把固定的非负的 v 个单位的内容从源头 s 运送到汇聚 t 的流，并以最小成本完成运送. 也就是说，我们要寻找一个流 $x=\{x_{ij}\}$，使得它的值为 v 且 $\sum a_{ij}x_{ij}$ 为最小. 我们称此为**最小成本流问题**(minimum-cost flow problem).

785

为了说明，考虑图 13.19 中的有向网络 a. 把方格中的数考虑为容量，圆圈中的数考虑为成本. 存在三个值为 3 的流；这些信息如图 13.26 所示. 其中的流 c 有最小成本，即

$$2\cdot3+2\cdot3+0\cdot0+1\cdot2+1\cdot2=16.$$

而流 a 和 b 的成本分别为 18 和 17.

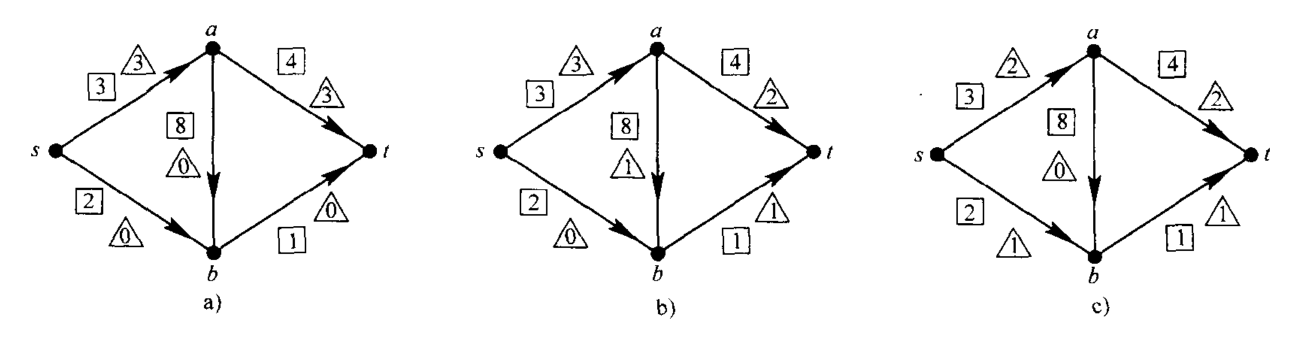


图 13.26 图 13.19 中的有向网络 a 的三个值为 3 的流，其容量如方格中数字所示，流如三角形中数字所示

【例 13.10 确定旅游团路径(Minieka[1978])】 一个 75 人的旅游团打算从纽约到火奴鲁鲁. 给该团安排的最少费用路径是什么？我们以城市作为有向网络的顶点，弧表示直接航线，每一条弧上有容量限制(非预订座位)和费用限制(航空费用)，构建一个有向网络. 为了求解该问题，我们在这个有向网络中寻找从纽约到火奴鲁鲁的 75 人最小成本流(这要假设航空费用是沿航线费用的简单和，而这一假设通常是不成立的). ■

786

【例 13.11 运输问题】 想象一下，我们把某种特定的日用品存储于 n 个仓库，并且要把它们运往 m 家超市. 设 a_i 是第 i 个仓库的日用品供应量，b_j 是第 j 家超市对此日用品的需求量，并设 a_{ij} 是从仓库 i 到超市 j 运输一个单位的日用品的成本. 为了简便，我们假设 $\sum a_i=\sum b_j$，即总供应量等于总需求量(很容易消除这一假设，参见练习 10). 这一问题是寻找最小化总运输成本的运送模式.

这一问题可以如下公式化. 设 x_{ij} 是从 i 到 j 被运送的日用品的单位数量. 我们设法最小化下式：

$$\sum_{\substack{i=1\\j=1}}^{\substack{m\\n}} a_{ij}x_{ij}$$

满足下面的限制条件：对于每一个 i 有

$$\sum_{j=1}^{m} x_{ij}\leqslant a_i, \tag{13.13}$$

且对于每一个 j，有

$$\sum_{i=1}^{n} x_{ij}\geqslant b_j. \tag{13.14}$$

限制条件(13.13)说明从第 i 个仓库运送的日用品总量至多等于那里的库存量，而限制条件

(13.14)说明运往第 j 家超市的日用品总量至少等于它的需求量. 注意因为 $\sum a_i=\sum b_j$，对于所有 i 和 j，满足(13.13)式和(13.14)式的任意解也将满足

$$\sum_{j=1}^{m} x_{ij} = a_i \tag{13.15}$$

和

$$\sum_{i=1}^{n} x_{ij} = b_j. \tag{13.16}$$

我们可以把这一运输问题看成是最小成本流问题. 画出一个有向图，其顶点为 n 个仓库 w_1，w_2，…，w_n，和 m 个超市 k_1，k_2，…，k_m. 加入源头 s 和汇聚 t，加入从每一个仓库到每一家超市的弧，从源头顶点到所有仓库的弧，以及从每一家超市到汇聚顶点的弧(参见图 13.27). 在弧(w_i，k_j)上放置成本 a_{ij} 和容量 $c_{ij}=\infty$(或非常大的数)；在弧(s，w_i)上放置成本 0 和容量 a_i；在弧(k_j，t)上放置成本 0 和容量 b_j. 因为我们有限制条件(13.13)和限制条件(13.14)，容量看到，对于这个有向网络，我们有最小成本流问题：我们寻找一个值等于 $\sum a_i=\sum b_j$ 的最小成本流. ■

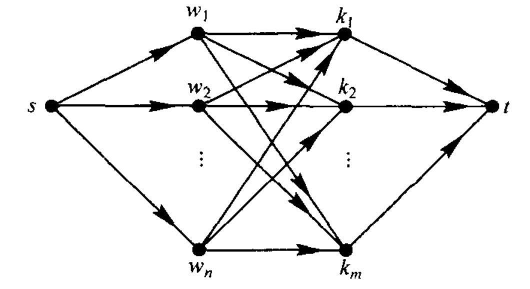


图 13.27 运输问题的有向网络

787

【例 13.12 最优分配问题(再探例 12.6)】 在例 12.6 及 12.7.2 节，我们讨论了下面的工作分配问题. 有 n 名工人和 m 项工作，每名工人适合每项工作，且对工人 i 做工作 j 的可能的能力由等级 r_{ij} 给出. 我们希望给工人分配工作，使得最大化能力等级的和. 设 x_{ij} 是一个变量，如果给工人 i 分配工作 j 则它等于 1，否则它等于 0. 这时，我们希望最大化下式：

$$\sum_{\substack{i=1\\j=1}}^{\substack{m\\n}} r_{ij}x_{ij},$$

其条件限制为

$$\sum_{i=1}^{n} x_{ij} \leqslant 1 \quad 和 \quad \sum_{j=1}^{m} x_{ij} \geqslant 1. \tag{13.17}$$

(注意，在 12.7.2 节中，我们取 $m=n$. 这里，我们考虑一般情况.)我们可以把这一问题考虑成为运输问题(例 13.11)，其方法是让工人对应于仓库，而让工作对应于超市. 这时，我们取所有 a_i 和 b_j 为 1 并设成本 $c_{ij}=-r_{ij}$. 我们设法最大化 $\sum_{i,j} r_{ij}x_{ij}$ 或最小化 $\sum_{i,j} c_{ij}x_{ij}$. 因此，这一优化分配问题也是最小成本流问题. [二者之间的一个不同点是我们有 $x_{ij}=0$ 或者 1 的附加要求，然而，可以证明：如果最小成本流问题有整数容量、成本和值 v，那么存在 x 为整数的最优解，而且标准算法产生整数解(见下面). 于是，(13.17)式的要求和 $x_{ij}\geqslant 0$ 的附加要求足以给出 $x_{ij}=0$ 或 1.] ■

【例 13.13 建筑物疏散(Ahuja，Magnanti 和 Orlin[1993]、Chalmet，Francis 和 Saunders[1982])】 2001 年 9 月 11 日的美国世贸中心倒塌引起人们对高层建筑物中疏散方法的关注. 这样的建筑物设计必须考虑到快速疏散. 建筑物的疏散模型对建筑物的设计阶段有帮助. 下面是经过简化的问题模型. 一个建筑物中有很多疏散人员的位置和很多出口. 为简单起见，我们假设只存在一个必须实施疏散的位置且仅有一个出口，这时，这个疏散口和出口就分别变成一个有向网络 N 的源头和汇聚. 假设我们必须从出口疏散的人的数量为 v. 一个建筑物有很多可以变成这个有向网络 N 的剩余顶点的位置. 当从位置 i 到 j 存在直接通道(楼梯或走廊)时，我们用一条弧连接顶点 i 到顶点 j，而且我们设弧(i，j)的容量 C_{ij} 是每单位时间从这个通道经过的人数. 注意经过通道(i，j)可能要花费若干单位时间 t_{ij}. 再一次简化这一问题，我们假设 t_{ij} 是整数. 现在，我们用

788

一个更大的有向网络 N'取代有向网络 N，在 N'中，我们对每一个顶点(包括源头和汇聚)做 C 次拷贝，其中 C 要选得足够大以确保我们能在 p 个单位时间内撤离这一建筑. 我们把顶点 i 的拷贝 i_p 看成是在时刻 p 时的该顶点. 在这个更大的有向网络 N'中，如果(i, j)在 N 中且 $q-p=t_{ij}$，那么我们画出一条从 i_p 到 j_q 的弧. 换句话说，弧(i_p, j_q)表示一个人沿着从 i 到 j 这条通道移动所化时间内顶点 i 到顶点 j 的移动. 在新弧(i_p, j_q)上放置容量 c_{ij}，和"成本"t_{ij}. 增加一个源头顶点 S 和汇聚顶点 T，并对于所有 p 加入弧(S, s_p)，对所有 q 加入弧(t_q, T)，其容量无限大，而成本是 0. 图 13.28 给出一个虚设小型建筑物的有向网络 N 和 N'. 从 S 到 T 的值为 v 的最小"成本"流给我们一个在最小的时间内从这个建筑物疏散所有 v 个人的计划. ■

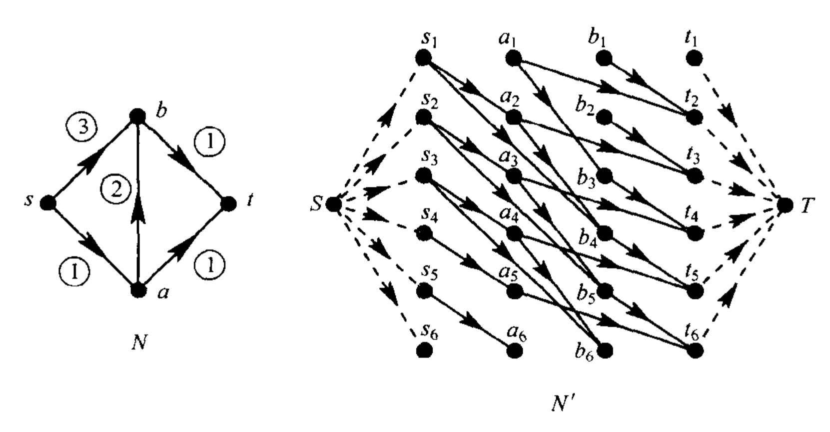


图 13.28 虚设小型建筑物的有向网络 N 和 N'，其中 $C=6$. 除虚线弧外，所有的容量都等于 2，而虚线弧的容量为无穷大(和 0 成本). N 中圆圈中的数是时间 t_{ij}. N'中弧(i_p, j_q)上的成本是 N 中弧(i, j)上的成本

789 通过修改 13.3.5 节中描述的最大流算法，我们得到解决最小成本流问题的有效算法(如果容量、成本和值 v 都是整数，则这一算法产生整数解). 其详细内容可以参见 Ahuja，Magnanti 和 Orlin[1993]、Lawler[1976]、Minieka[1978]或 Papadimitriou 和 Steiglitz[1982]. 对于诸如运输问题或优化分配问题的特殊情况，存在更有效的算法.

本节练习

1. 在图 13.20 中的每一个有向网络中，把圆圈中的数看作成本而不是流，把方格中的数如以往一样看作是容量. 对于每一种情况，寻找从 s 到 t 的值为 3 的最小成本流.
2. 在图 13.29 中的每一个有向网络中，把每一条弧(i, j)上的序对(a_{ij}, c_{ij})考虑为它的成本(a_{ij})和容量(c_{ij}).
   (a)寻找有向网络 a 中从 s 到 t 的值为 25 的最小成本流.
   (b)在有向网络 b 中寻找从 s 到 t 的值为 9 的最小成本流.
   (c)在有向网络 c 中，寻找从 s 到 t 的值为 30 的最小成本流.

790

3. (a)对于图 13.28(例 13.13)所示的建筑物疏散问题，寻找值为 10 的最小成本流.
   (b)通过如何从图 13.28 所描述的建筑物中疏散 10 个人来说明(a)的答案.
   (c)可以从图 13.28 所描述的建筑物中疏散的人的最大数量是多少？这个值的最小成本流是什么？
4. (Ahuja，Magnanti 和 Orlin[1992])一家航空公司从纽约到芝加哥一天有 6 次航班. 从早晨 7 点开始，它们每隔 2 小时间起飞一次. 上午航班的飞机有 100 名的客容量. 午后的飞机有 150 名的客容量. 这一航空公司可以把预定过多的乘客转载到后面的航班. 下午 5 点之后，这家航空公司可以把乘客转载到另一家航空公司的晚间 10 点的航班，那个航班总有座位. 被转载的乘客可以得到 200 美元的补偿，而且如果航班延迟的时间超过 2 小时，那么每小时再加 20 美元补偿. 假设在某一天，在它的六次航班上，这家航空公司分别有 110，160，103，149，175，140 人定票. 把最小化补偿成本问题阐述为最小成本流问题.
5. 一名酒宴承办人预先知道在以后的 n 天中，在第 j 天需要 a_j 条餐巾，其中 $j=1, 2, \cdots, n$. 对于任意给

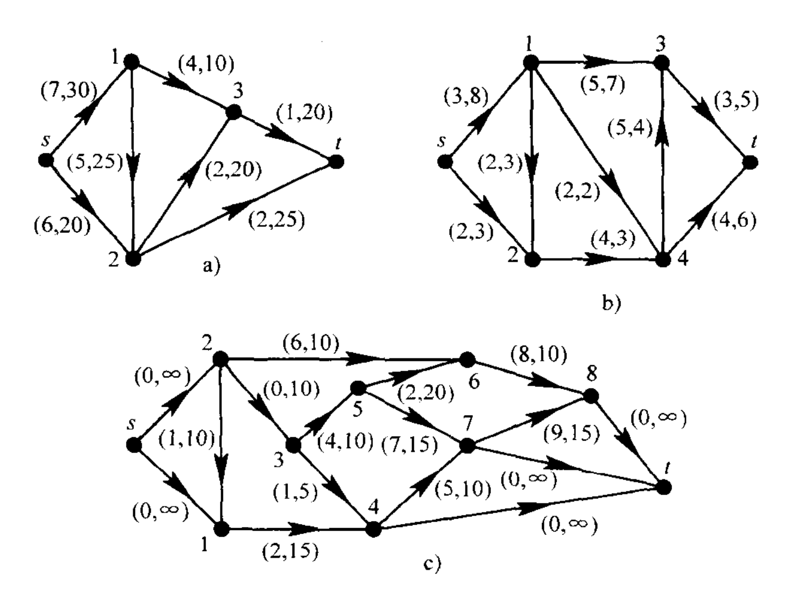


图 13.29　练习 2 的最小成本流问题

定的某天，这名承办人可能买新的餐巾或使用洗过的餐巾．餐巾的洗涤可以由快速服务店来做，它需要 q 天，或较慢的服务店来做，则需要 r 天．新的餐巾的成本是 c 分钱，快速洗涤一条餐巾的成本是 d 分钱，而慢速洗涤一条餐巾的成本是 e 分钱．从没有餐巾开始，这名承办人如何以最小的成本满足餐巾的需求？把这一问题转化成最小流成本问题(注意，在历史上早于这一问题的一个类似问题是，维护飞机时，快速检查还是慢速检查发动机的问题)．

6. (Ahuja，Magnanti 和 Orlin[1993]、Prager[1957])一名投资者考虑在 T 个期货时间周期内投资黄金．在每一个时间周期内，他可以买进、卖出或持有他已有的黄金．假设在时间周期 i 内，他可以至多买进 α_i 盎司黄金，持有至多 β_i 盎司黄金，而且(因为先前的合同)必须至少卖出 γ_i 盎司黄金，而且假设直到下一个时间周期之前，他必须至少持有买进的黄金．在时间周期 i 内投资黄金的相关成本是，每盎司买进价为 p_i、卖出价为 s_i，及在时间周期 i 内持有黄金的成本是每盎司 w_i．这位投资者在 T 个时间周期内如何买进、卖出和持有黄金才能最大化利润？把这一问题阐述为最小成本流问题．
7. 在练习 6 的投资问题中，假设我们认为各种限制和成本与 i 无关，但是如果你买进黄金，那么你必须提前两个时间周期订购，而且如果你决定在任意时间周期内持有黄金，那么你必须持有它三个时间周期．你如何分析这一投资问题？[提示：与例 13.13 的建筑物疏散比较]．
8. (Cook 等[1998])假设 D 是弱连通有向图．我们可以以成本 $c(a)$“复制”D 的任意弧 $a=(u, v)$，其中复制的意思是加入另一条从 u 到 v 的弧．我们可以复制一条弧任意多次．我们想尽可能少地复制弧以得到一个有欧拉封闭路径的多重有向图．把这一问题阐述为最小成本流问题(类似的问题出现于与 12.7.1 节的“中国邮差”问题相关的问题中)．

791

9. 假设我们有两个仓库和两家超市．在第一个仓库有 10 卷电线，在第二个仓库有 14 卷电线．另外，第一家超市需要 13 卷电线，而第二家需要 11 卷电线．当运输成本由下面的矩阵给出时，寻找最小成本运输调度．

		工厂 1	工厂 2
仓库	1	100	84
	2	69	75

10. 考虑运输问题(例 13.11)．

(a)证明如果 $\sum a_i < \sum b_j$，那么不存在满足(13.13)式和(13.14)式的运送模式．

(b)如果 $\sum a_i > \sum b_j$，证明我们还可以通过下面的方法认为 $\sum a_i = \sum b_j$：创建一个新的第 $(m+1)$ 个超市，设 $b_{m+1} = \sum_{i=1}^{n} a_i - \sum_{j=1}^{m} b_j$，加入从每一个仓库到新超市的弧，并设所有新弧的容量为 0.

11. 在有成本的有向网络中，修改增广链的定义，使得它可以开始于 s 之外且结束于 t 之外的顶点. 定义增广链的成本为向前弧的成本和减去向后弧的成本和. *增广回路*(augmenting circuit)是形成回路的增广链.
 (a)对于练习 1 中的每一个例子，寻找值为 3 的没有最小成本的流，并寻找有负成本的流增广回路.
 (b)证明：如果值为 v 的流有最小成本，那么它没有负成本流增广回路.
 (c)证明：如果值为 v 的流没有负成本流增广回路，那么这个流有最小成本.
12. 我们可以把运输问题(例 13.11)如下简化成最大权匹配问题(12.1 节). 如下构建一个有 $2\sum a_i$ 个顶点的二部图$G=(X, Y, E)$. 对于$i=1, 2, \cdots, n$, 设 X 是由第 i 个仓库的a_i 个拷贝组成的，而对于 $j=1, 2, \cdots, m$, 设 Y 是由第 j 个超市的 b_j 个拷贝组成的. G 包含 X 中顶点与 Y 中顶点间的所有可能边. 在将第 i 个仓库的拷贝和第 j 个超市的拷贝连接起来的边上，放置权 $K-a_{ij}$，其中 K 足够大. 证明运输问题的最优解对应于 G 的最大权匹配.

参考文献

AHUJA, R. K., MAGNANTI, T. L., and ORLIN, J. B., "Network Flows," in G. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd (eds.), *Handbooks in Operations Research and Management Science*, Vol. 1: *Optimization*, North-Holland, Amsterdam, 1989, 211–369.

AHUJA, R. K., MAGNANTI, T. L., and ORLIN, J. B., "Some Recent Advances in Network Flows," *SIAM Review, 33* (1991), 175–219.

792

AHUJA, R. K., MAGNANTI, T. L., and ORLIN, J. B., *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, NJ, 1993.

ARONSON, J. E., "A Survey of Dynamic Network Flows," *Ann. Oper. Res., 20* (1989), 1–66.

ARRATIA, R., and LANDER, E. S., "The Distribution of Clusters in Random Graphs," *Adv. Appl. Math., 11* (1990), 36–48.

BARITCHI, A., COOK, D. J., and HOLDER, L. B., "Discovering Structural Patterns in Telecommunications Data," in *Proceedings of the Thirteenth Annual Florida AI Research Symposium*, 2000.

BAZARAA, M. S., JARVIS, J. J., and SHERALI, H. D., *Linear Programming and Network Flows*, 2nd ed., Wiley, New York, 1990.

BERGE, C., and GHOUILA-HOURI, A., *Programming, Games and Transportation Networks*, Wiley, New York, 1965.

BONDY, J. A., and MURTY, U. S. R., *Graph Theory with Applications*, American Elsevier, New York, 1976.

BORŮVKA, O., "O Jistém Problému Minimálním," *Práce Mor. Přírodoved. Spol. v Brne (Acta Soc. Sci. Nat. Moravicae), 3* (1926), 37–58. (a)

BORŮVKA, O., "Příspevěk k. Řešení Otázky Ekonomické Stavby Elektrovodních Sítí," *Elektrotech. Obzor, 15* (1926), 153–154. (b)

CHALMET, L. G., FRANCIS, R. L., and SAUNDERS, P. B., "Network Models for Building Evacuation," *Management Sci., 28* (1982), 86–105.

CHERITON, D., and TARJAN, R. E., "Finding Minimum Spanning Trees," *SIAM J. Comput., 5* (1976), 724–742.

CHURCH, K., "Massive Data Sets and Graph Algorithms in Telecommunications Systems," Session on Mathematical, Statistical, and Algorithmic Problems of Very Large Data Sets, American Mathematical Society Meeting, San Diego, CA, January 1997.

COOK, W. J., CUNNINGHAM, W. H., PULLEYBLANK, W. R., and SCHRIJVER, A., *Combinatorial Optimization*, Wiley, New York, 1998.

CZEKANOWSKI, J., "Zur Differentialdiagnose der Neandertalgruppe," *Korrespondenzbl. Dtsch. Ges. Anthrop., Ethn. Urg., 40* (1909), 44–47.

CZEKANOWSKI, J., "Objektive Kriterien in der Ethnologie," *Ebenda, 43* (1911), 71–75.

CZEKANOWSKI, J., "Das Typenfrequenzgesetz," *Anthropol. Anz., 5* (1928), 15–20.

DIJKSTRA, E. W., "A Note on Two Problems in Connexion with Graphs," *Numer. Math., 1* (1959), 269–271.

EDMONDS, J., and KARP, R. M., "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems," *J. ACM, 19* (1972), 248–264.

ELIAS, P., FEINSTEIN, A., and SHANNON, C. E., "Note on Maximum Flow through a Network," *IRE Trans. Inf. Theory, IT-2* (1956), 117–119.

FORD, L. R., and FULKERSON, D. R., "Maximal Flow through a Network," *Canad. J. Math., 8* (1956), 399–404.

FORD, L. R., and FULKERSON, D. R., "A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problem," *Canad. J. Math., 9* (1957), 210–218.

FORD, L. R., and FULKERSON, D. R., *Flows in Networks*, Princeton University Press, Princeton, NJ, 1962.

FRANK, C. R., "A Note on the Assortment Problem," *Management Science, 11* (1965), 724–726.

FRANK, H., and FRISCH, I. T., *Communication, Transportation and Flow Networks*, Addison-Wesley, Reading, MA, 1971. [793]

GABOW, H. N., GALIL, Z., SPENCER, T., and TARJAN, R. E., "Efficient Algorithms for Finding Minimum Spanning Trees in Undirected and Directed Graphs," *Combinatorics, 6* (1986), 109–122.

GLOVER, F., and KLINGMAN, D., "Network Applications in Industry and Government," *AIIE Trans., 9* (1977), 363–376.

GODEHART, E., *Graphs as Structural Models*, 2nd ed., Friedr. Vieweg & Sohn, Braunschweig, Germany, 1990.

GOLDMAN, A. J., "Discrete Mathematics in Government," lecture presented at SIAM Symposium on Applications of Discrete Mathematics, Troy, NY, June 1981.

GONDRAN, M., and MINOUX, M., *Graphs and Algorithms*, Wiley, New York, 1984.

GOWER, J. C., and ROSS, G. J. S., "Minimum Spanning Trees and Single Linkage Cluster Cluster Analysis," *Applied Statistics, 18*, (1969), 54–64.

GRAHAM, R. L., and HELL, P., "On the History of the Minimum Spanning Tree Problem," *Annals of the History of Computing, 7* (1985), 43–57.

GUÉNOCHE, A., HANSEN, P., and JAUMARD, B., "Efficient Algorithms for Divisive Hierarchical Clustering with the Diameter Criterion," *J. Classification, 8* (1991), 5–30.

HANSEN, P., FRANK, O., and JAUMARD, B., "Maximum Sum of Splits Clustering," *J. Classification, 6* (1989), 177–193.

HANSEN, P., and JAUMARD, B., "Minimum Sum of Diameters Clustering," *J. Classification, 4* (1987), 215-226.

HANSEN, P., JAUMARD, B., and MLADENOVIC, N., "Minimum Sum of Squares Clustering in a Low Dimensional Space," *J. Classification, 15* (1998), 37–55.

HU, T. C., *Integer Programming and Network Flows*, Addison-Wesley, Reading, MA, 1969.

IRI, M., *Network Flows, Transportation and Scheduling*, Academic Press, New York, 1969.

JARNÍK, V., "O Jistém Problému Minimálním," *Práce Mor. Přírodoved Spol. v Brne (Acta Soc. Sci. Nat. Moravicae), 6* (1930), 57–63.

JOHNSON, D. B., "Priority Queues with Update and Minimum Spanning Trees," *Inf. Process. Lett., 4* (1975), 53–57.

KARGER, D., "Information Retrieval: Challenges in Interactive-Time Manipulation of Massive Text Collections," Session on Mathematical, Statistical, and Algorithmic Problems of Very Large Data Sets, American Mathematical Society Meeting, San Diego, CA, January 1997.

KERSHENBAUM, A., and VAN SLYKE, R., Computing Minimum Spanning Trees Efficiently, ACM 72, *Proceedings of the Annual ACM Conference*, 1972, 518–527.

KRUSKAL, J. B., "On the Shortest Spanning Tree of a Graph and the Traveling Salesman Problem," *Proc. Amer. Math. Soc., 7* (1956), 48–50.

LAWLER, E. L., *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.

MENGER, K., "Zur allgemeinen Kurventheorie," *Fund. Math., 10* (1927), 96–115.

MINIEKA, E., *Optimization Algorithms for Networks and Graphs*, Dekker, New York, 1978.

MIRKIN, B., *Mathematical Classification and Clustering*, Kluwer Academic Publishers, Dordrecht-Boston-London, 1996.

NEIL, M. D., "Multivariate Assessment of Software Products," *Software Testing, Verification & Reliability, 1* (1992), 17–37.

794 NEŠETŘIL, J., MILKOVÁ, E., and NEŠETŘILOVÁ, H., "Otakar Borůvka on Minimum Spanning Tree Problem: Translation of Both the 1926 Papers, Comments, History," *Discrete Math., 233* (2001), 3–36.

PAPADIMITRIOU, C. H., and STEIGLITZ, K., *Combinatorial Optimization: Algorithms and Complexity*, Prentice Hall, Englewood Cliffs, NJ, 1982.

PRAGER, W., "On Warehousing Problems," *Oper. Res., 5* (1957), 504–512.

PRIM, R. C., "Shortest Connection Networks and Some Generalizations," *Bell Syst. Tech. J., 36* (1957), 1389–1401.

RAVINDRAN, A., "On Compact Book Storage in Libraries," *Opsearch, 8* (1971), 245–252.

STONE, H. S., "Multiprocessor Scheduling with the Aid of Network Flow Algorithms," *IEEE Trans. Software Engrg., 3* (1977), 85–93.

TARJAN, R. E., *Data Structures and Network Algorithms*, SIAM, Philadelphia, 1983.

TARJAN, R. E., "A Simple Version of Karzanov's Blocking Flow Algorithm," *Oper. Res. Lett., 2* (1984), 265–268. (a)

TARJAN, R. E., "Efficient Algorithms for Network Optimization," *Proceedings of the International Congress of Mathematicians*, (Warsaw, 1983) Vols. 1, 2 (1984), 1619–1635. (b)

WHITE, L. S., "Shortest Route Models for the Allocation of Inspection Effort on a Production Line," *Management Sci., 15* (1969), 249–259.

YAO, A., "An $O(|E| \log\log|V|)$ Algorithm for Finding Minimum Spanning Trees," *Information Processing Letters, 4* (1975), 21–23.

795 ZAHN, C. T., "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters," *IEEE Trans. Computing, C-20* (1971), 68–86.

人名索引[⊖]

索引中的页码为英文原书页码，与书中页边标注的页码一致.

⊖ 斜体页码是指每章末尾参考文献的页码.

主题索引

索引中的页码为英文原书页码，与书中页边标注的页码一致.

B

D

E

F

G

H

I

N

O

P

Q

R

S

T

U

V

W

X

Y

Z